SPEECH IMITATION BY NEURAL SPEECH SYNTHESIS WITH ON-THE-FLY DATA AUGMENTATION

by

CHUNG MAN HON

A Thesis Submitted to
The Hong Kong University of Science and Technology
in Partial Fulfillment of the Requirements for
the Degree of Master of Philosophy
in Computer Science and Engineering

January 2022, Hong Kong

Authorization

I hereby declare that I am the sole author of the thesis.

I authorize the Hong Kong University of Science and Technology to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize the Hong Kong University of Science and Technology to reproduce the thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

_____

CHUNG MAN HON

7 January 2022

SPEECH IMITATION BY NEURAL SPEECH SYNTHESIS WITH ON-THE-FLY DATA AUGMENTATION

by

CHUNG MAN HON

This is to certify that I have examined the above M.Phil. thesis
and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by
the thesis examination committee have been made.

Prof. Brian Mak, Thesis Supervisor

Prof. Dit-Yan Yeung, Head of Department

Department of Computer Science and Engineering

7 January 2022

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my supervisor, Prof. Brian Mak, for his supportive guidance and invaluable advice during my postgraduate study. I acquired a better skill in developing a research idea analytically and presenting the finding in a persuasive approach.

I am also very appreciative to my supervisor at work, Dr. C.D. Shum, who encouraged me to study further and introduced me to Prof. Mak.

I would also like to thank the research team members at HKUST, Liu Zhaoyu, Zhu YingKe, Niu Zhe, and colleagues at LSCM R&D Centre, Cesar Mak, Amy He, Insu Song, for their sharing of knowledge and help.

Last but not least, I am thankful to my family for their support and encouragement throughout my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SPEECH IMITATION BY NEURAL SPEECH SYNTHESIS WITH ON-THE-FLY DATA AUGMENTATION

by

CHUNG MAN HON

Department of Computer Science and Engineering

The Hong Kong University of Science and Technology

ABSTRACT

Recent deep learning text-to-speech (TTS) systems synthesize natural speech. Applying speaker adaptation can make a TTS system speak like the adapting speaker, but the speaking style of the synthesized utterance still follows closely to the speaker's of the training utterances. In some applications, it is desirable to synthesize speech in a speaking manner depending on the scenario. A straightforward solution is to record speech data from a speaker under different role-playing scenarios. However, excluding professional voice talents, most people are not experienced in speaking in different expressive styles. Likewise, without being exposed to a multilingual environment from an early age, most people cannot speak a second language with its native accent. In this thesis, we propose a novel data augmentation method to create a stylish TTS model for a speaker. Specifically, augmented data are created by 'forcing' a speaker to imitate stylish speeches of other speakers. Our proposed method consists of two steps. Firstly, all the data are used to train a basic multi-style multi-speaker TTS model. Secondly, augmented utterances are created on-the-fly from the latest TTS model during its training and are used to fur-

ther train the TTS model. We select two applications to demonstrate the effectiveness of our proposed method: (1) synthesizing speech in three scenarios — newscasting, public speaking, and storytelling — for a speaker who provides only neutral speech data; (2) synthesizing "beautified" speech of a language spoken by a non-native speaker by reducing his/her accent in the aspects of better pronunciation and more native prosody. Our experiment shows that for scenario-based TTS, the scenario speeches synthesized by our proposed method are overwhelmingly preferred over those from a speaker-adapted TTS model. For accent-beautified TTS, our model reduces the foreign accent of the non-native speeches while retaining a higher voice similarity than a state-of-the-art accent conversion model.

CHAPTER 1

INTRODUCTION

Text-to-speech (TTS) system has a long research history. To understand what a TTS system does, we may refer to a well-known example. Stephen Hawking [54] was a renowned theoretical physicist from England. He suffered from the motor neuron disease known as amyotrophic lateral sclerosis (ALS). He lost his speaking ability because of a medical condition in 1985. Since then, he relied on a TTS voice assistive machine to speak for him. The machine consisted of a computer and a voice synthesizer. The computer provided a running list of words. Hawking twitched his cheek muscle to control a cursor to select a word or phrase from the list. Then he repeated the process to pick the next word or phrase until he created a sentence. Finally, the voice synthesizer read the sentence aloud for him. This voice spoke in a robotic tone under any scenario, but for a human's utterance, the speaking style of a speech is scenario-dependent. Moreover, contradicting his British identity, the voice used in his synthesizer had an American accent.

1.1 Targeted Speech Synthesis Applications

Speech is a form of verbal communication, which allows us to communicate efficiently. Inevitably, the nonverbal channel of communication, like the tone and intensity of voice, plays a key role in an individual's perception regarding a speech, such as politeness [23]. In public speaking, one particular technique for the speaker to keep the audience engaged is to bring the pitch up a notch when talking about an interesting part of the speech [39]. People could acquire better speaking skills via a "practice makes perfect" approach.

Everyone knows speech imitation. An infant starts his/her speech development by matching the caretaker's speech in their interaction [32]. A professional voice actor could even imitate the voice identity of another speaker. In this research work, we would like to create a customized TTS voice with a speaking style different from the original speaker's

utterances without altering the speaker's voice identity. We list out two applications below.

1.1.1 Scenario-based Speech Synthesis for a Neutral Speaker

With the rapid advancement of artificial intelligence, TTS technology provides a very natural artificial voice nowadays. Consequently, its application has gained popularity in our daily lives. For instance, Apple's iPhone has a voice assistant named Siri while Amazon's smart home speaker has Alexa. Both use a TTS system to communicate with humans. It recognizes your questions in speech. It searches the information online and reads the answer aloud. Their voice shall be in a conversational style rather than talking constantly. One user made an interesting attempt. He asked the voice assistants to tell a joke. They told a good one in a flat tone [44]. The publication business industry also showed interest in this technology. They used a TTS system to read the book content of a newly published book. Then they organized the generated utterances into an audiobook [15]. The complete audio track for a typical book is usually a few hours long. Hence the artificial voice for this application shall be expressive enough to keep the reader engaged. Some start-ups [4, 31] further extended this idea. They provide easy integration of TTS to read articles written by a content provider. By converting a web article into a speech or even a podcast, this increases the reachability and accessibility of the content. In the future, we foresee that the author could use his/her voice in reading his/her articles or even books aloud. However, most common people are neutral speakers who speak with a neutral tone. On the other hand, the content of an article or book can be very dynamic, the synthesized speech has to be scenario-dependent even if the author is not a voice-over professional.

1.1.2 Accent-beautified Speech Synthesis for a Non-native Speaker

TTS may be applied to reduce the communication gap between people speaking in different mother tongues. The world is more connected than ever before with travelers visiting different countries and businesspersons having online meetings with colleagues or clients from different countries. In the communication between a native speaker of a language and a non-native speaker speaking the same language, the intelligibility and fluency of

the foreign speech are the key factors [17] for understanding. There are a few solutions to facilitate this type of communication. Accent conversion (AC) converts a non-native accented utterance from a second-language (L2) speaker to one with a more native accent without changing the speaker's voice. This technology can be used in language learning so that an L2 learner may listen to the difference between his/her imitated speech and the native speech to improve his/her L2 speaking skills [49]. Another technology is called simultaneous interpretation (SI). Some online meeting applications, e.g., Skype [41], notice that a foreign language and/or the accent of a speaker may affect the reception of his/her presentation from the audience, so it provides this SI feature. The feature allows the audience to listen in their first language (L1). It involves three steps: an L2 speech-to-text engine, a machine translation between L2 and L1, and an L1 text-to-speech. However, the voice identity of the synthesized speech is not the original speaker. Likewise, if a content provider writes his/her blog in a foreign language, it is desirable for the synthesized foreign speech based on his/her voice to sound like the speech from a native speaker of the language.

1.2 The Challenge

In many cases, people are looking for an artificial voice that is not only natural but also a customized and expressive voice. A customized voice sounds as similar as possible to the user's own voice. An expressive voice uses the right prosody to read a sentence for a wanted scenario.

To have synthesized speech sounds similar to the source speaker, recently advanced text-to-speech (TTS) technologies allow voice cloning through speaker adaptation. But it would carry the monotonous and non-native accent of the speaker. To create a multi-scenario synthetic voice, it is plausible to record high-quality audio training data by a voice talent who can manage to speak with different styles or languages natively. However, this only works for voice talents; most ordinary people are not able to utter in all wanted scenarios or accents. Similarly, linguists also point out that, unless exposed to a multilingual environment from an early age, most people could only speak a limited set of languages with a native accent.

1.3 Proposed Method

We propose a novel data augmentation framework to train a multi-style TTS model for a common person. Firstly, we collect some amount of his/her speech spoken in a neutral voice. Then we generate augmented data from the person speaking in wanted styles on-the-fly. Our idea is simple: a person learns to speak by imitating other people's speech. We integrate such behaviour in the TTS model training. In Tacotron2 [40], the attention alignment encapsulates the prosody of the generated utterance. The augmented data from the neutral speaker should follow the same prosody that a speaker speaks in the target style. With the additional use of scenario embedding computed from trained global style tokens [2, 52], our proposed TTS model could generate speeches that match the speaking prosody of the target scenarios. This proposed method is exemplified by two TTS models. One is for scenario-based speech synthesis for a neutral speaker. Another one is for accent-beautified speech synthesis for a non-native speaker.

Our main contributions are as follows.

- We propose a methodology for on-the-fly data augmentation in TTS training. The augmented data are constructed with the desired properties and are used as additional training samples.
- The stylish augmented training data can be generated from a neutral speaker by imitating the attention matrix of the stylish speech from another speaker on-the-fly.
- We successfully demonstrated the effectiveness of our proposed method on two applications. (1) We synthesize scenario-specific utterances with a neutral voice; (2) We synthesize accent-beautified utterances of L2 speeches for a non-native speaker.

The remaining parts of the thesis are organized as follows.

Chapter 2 provides some background on the TTS system. We review the classic speech synthesis pipeline, then the recent neural TTS model and its derivations for scenario-based TTS and accent conversion applications. We also review the TTS works that utilized data augmentation. Chapter 3 presents our proposed framework. It leads to the design of two

TTS models: a scenario-based TTS model and an accent-beautified TTS model. In Chapter 4, we report our experimental setup and evaluate the synthesized speech produced by our proposed models using both objective and subjective metrics. Finally, we conclude the thesis in Chapter 5.

CHAPTER 2

PRELIMINARIES AND RELATED WORK

In this chapter, we introduce the classical text-to-speech synthesis pipeline. Then we present the transition towards the neural speech synthesis model and the variants for stylish speech synthesis. Finally, we review data augmentation methods applied in TTS.

2.1 Basics of a Text-to-Speech (TTS) System

The classical way to implement a TTS system is to divide the system into a frontend and a backend module.

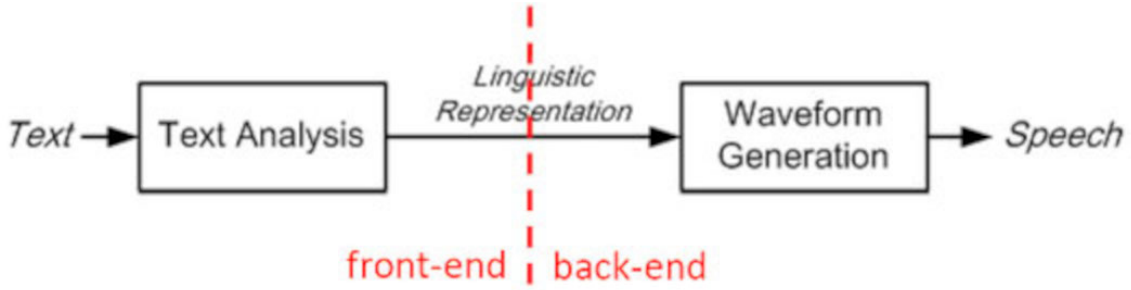

Figure 2.1. TTS components illustrated from [62].

2.1.1 Frontend

The frontend converts the input text into a linguistic representation. The input text is usually a sentence. The frontend consists of text analysis components, including text normalization and phonetization. It was usually implemented by a rule-based approach [63, 33], and more recently by a neural model [58, 59].

Text normalization expands the input tokens into words in the spoken language. Thereby numerical symbols and abbreviations are converted into their normal orthographic form.

Phonetization converts the orthographic form into phonemes, which are the smallest units of sound. Some illustrative examples are: "56 in. long" is read as "fifty-six inches long"; "7-11 convenience store" is read as "Seven-Eleven convenience store"; The "lives" in "it lives" and "nine lives" are read differently. Based on the sentence context, the pronunciation ambiguity of a word is resolved. Then a pronunciation dictionary is used to convert each word into phonemes. The CMU dictionary [46] consists of the pronunciations of over 134,000 English words. Its phoneme set is based on the ARPAbet symbols, which are shown in Table 2.1. It is suitable for phonetization. For example, the phonetic sequence of "it lives" is "IH1 T L IH1 V Z". Finally, the frontend returns a phonetic sequence as the linguistic representation of the input text.

ARPAbet phoneme set									
AA	D	IH	OW	UH	AE	DH	IY	OY	UW
AH	EH	JH	P	V	AO	ER	K	R	W
AW	EY	L	S	Y	AY	F	M	SH	Z
B	G	N	T	ZH	CH	HH	NG	TH	

Table 2.1. ARPAbet phoneme set.

2.1.2 Backend

The backend converts the linguistic representation into speech. The speech uttered by a human is a continuous waveform in the time domain. In the digital world, the waveform is sampled and encoded as a numerical sequence. For example, a CD soundtrack contains 44,100 samples per second.

A straightforward solution is to combine the audio clips of individual phonemes one by one to create a speech waveform. This idea leads to the development of concatenative speech synthesis. In this approach, a large database of speech segments is prepared. It consists of a single speaker reading a large amount of scripts. The scripts are designed such that they consist of popular words and phrases. During the synthesis of an unseen sentence, speech fragments are selected from the speech database according to its contents. The fragments are then concatenated to form a waveform of the speech of the target

sentence. Depending on the available database entries, there can be several possible sequence combinations. Various models [13, 3] have been designed to select the best one that minimizes the acoustic artifacts in the concatenated waveform. Inevitably, the naturalness of the synthesized speech depends heavily on the database coverage.

More often than not, the direct conversion from the relatively short phonetic sequence to the long digital audio is challenging. An intermediate representation is commonly engineered to break up the problem into multiple sub-problems. For instance, the time signal is transformed to frequency-domain. Then the original problem becomes two sub-problems and is to be solved by two models. Firstly, an acoustic model is designed to infer the characteristics of the frequency-domain signal from the phonetic sequence. Secondly, a vocoder model is trained to generate the time-domain waveform from the frequency-domain parameters.

2.2 Toward a Naturally Neural TTS

Naturalness refers to how natural a speech sounds. It was one of the key research objectives in the development of a TTS system for a long time [29, 40]. Statistical Parametric Speech Synthesis (SPSS) was once the state-of-the-art TTS methodology. After the breakthrough of deep learning [11], the research paradigm shifted from the SPSS approach to the neural speech synthesis approach [1, 40].

2.2.1 Statistical Parametric Speech Synthesis

Statistical Parametric Speech Synthesis (SPSS) comprises some models to predict the handcrafted acoustic features of the speech signal from the linguistic representation. Based on those acoustic features, we could reconstruct the speech signal via a vocoder model. We elaborate the concept by the classical hidden Markov model (HMM) based TTS [57] and STRAIGHT vocoder [19].

Classical Vocoder

Humans produce sound when air passes through the vocal cords. If the vocal cords vibrate, it is a voiced sound. Otherwise, it is an unvoiced sound. The complex process is modeled as a source-filter model shown in Fig. 2.2. The STRAIGHT vocoder receives a pulse train signal. It modifies the pulse according to the acoustic features extracted from the speech. For example, the fundamental frequency F0 is the frequency at which vocal cords vibrate in voiced sounds [25]. Hence it is an input parameter to a periodic pulse generator.

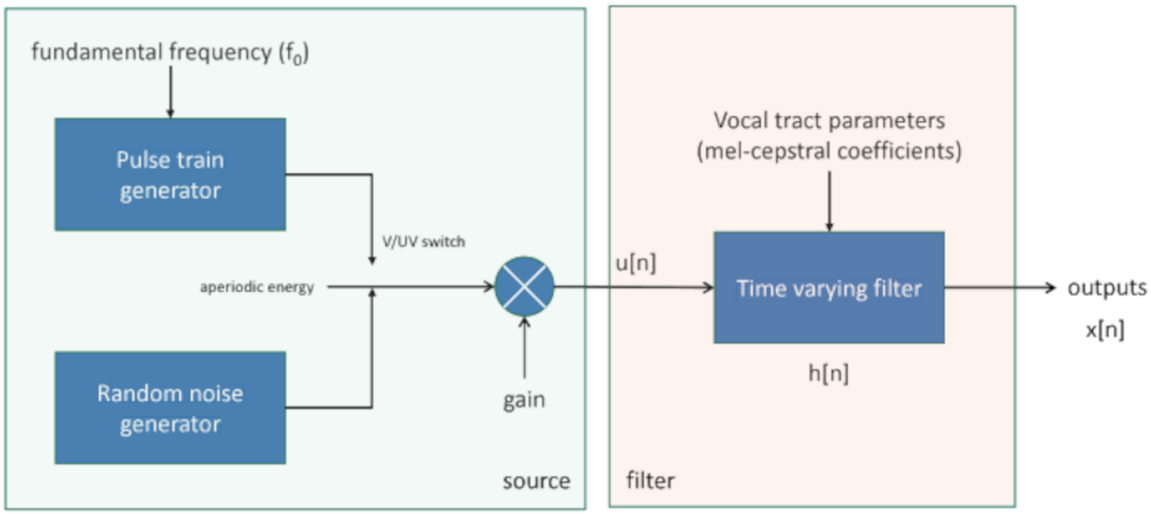

Figure 2.2. Source-filter model. The figure is extracted from [7].

Classical Acoustic Model

HMM is used to generate acoustic features from a phonetic sequence. Firstly, the acoustic features are extracted from the training audios. The feature at each timestamp is aligned to each phoneme in the phonetic sequence. Consequently, there are multiple instances of observed acoustic features for each phoneme. Then HMM is used to model the context-dependent distribution of the acoustic features of individual phonemes. Each HMM is further associated with a state duration density. During speech synthesis of a sentence, HMMs of phonemes in the phonetic sequence of the sentence are concatenated. The output acoustic features sequence from the resulting sentence-level context-dependent HMM

is then passed to the vocoder for waveform generation. The detailed process could be referred to Fig. 2.3.

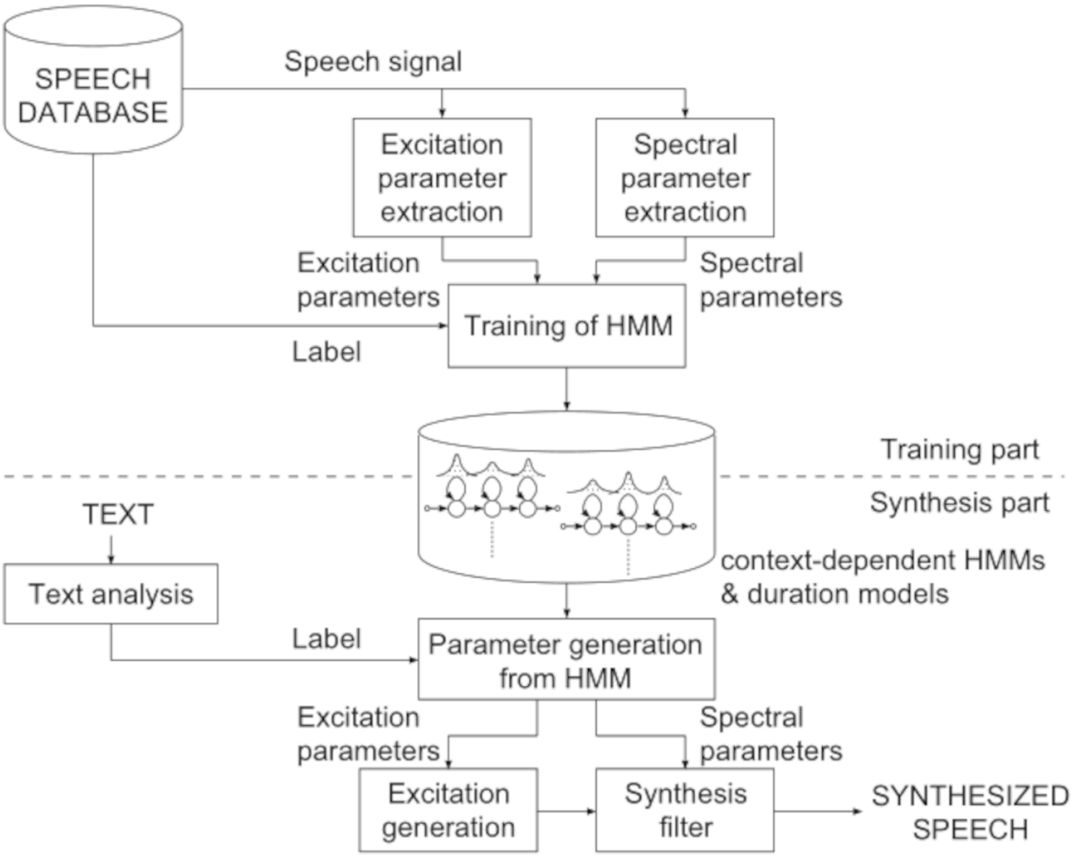

Figure 2.3. A typical HMM-based TTS. The figure is extracted from [57].

Shortcoming

SPSS could not synthesize speech as natural as human speech [10]. SPSS requires handcrafted acoustic feature selection. The domain-specific expertise has to design and tune the system. Still, those features are a lossy representation of the audio signal. Furthermore, the HMMs's accuracy depends on the precise alignment between phonemes and their corresponding waveforms and the available amount of training data. HMMs also tend to over-smooth the detail of the original features. Therefore, the reconstruction of speech by SPSS contains artifacts and is not natural enough.

2.2.2 Neural Speech Synthesis

Tacotron2 [40] is a text-to-spectrogram prediction model. WaveNet [48] is a vocoder that converts a spectrogram into a waveform. Both of them are deep learning-based models. By utilizing the two models together [40], the synthesized speech reaches a similar naturalness to professionally recorded speech.

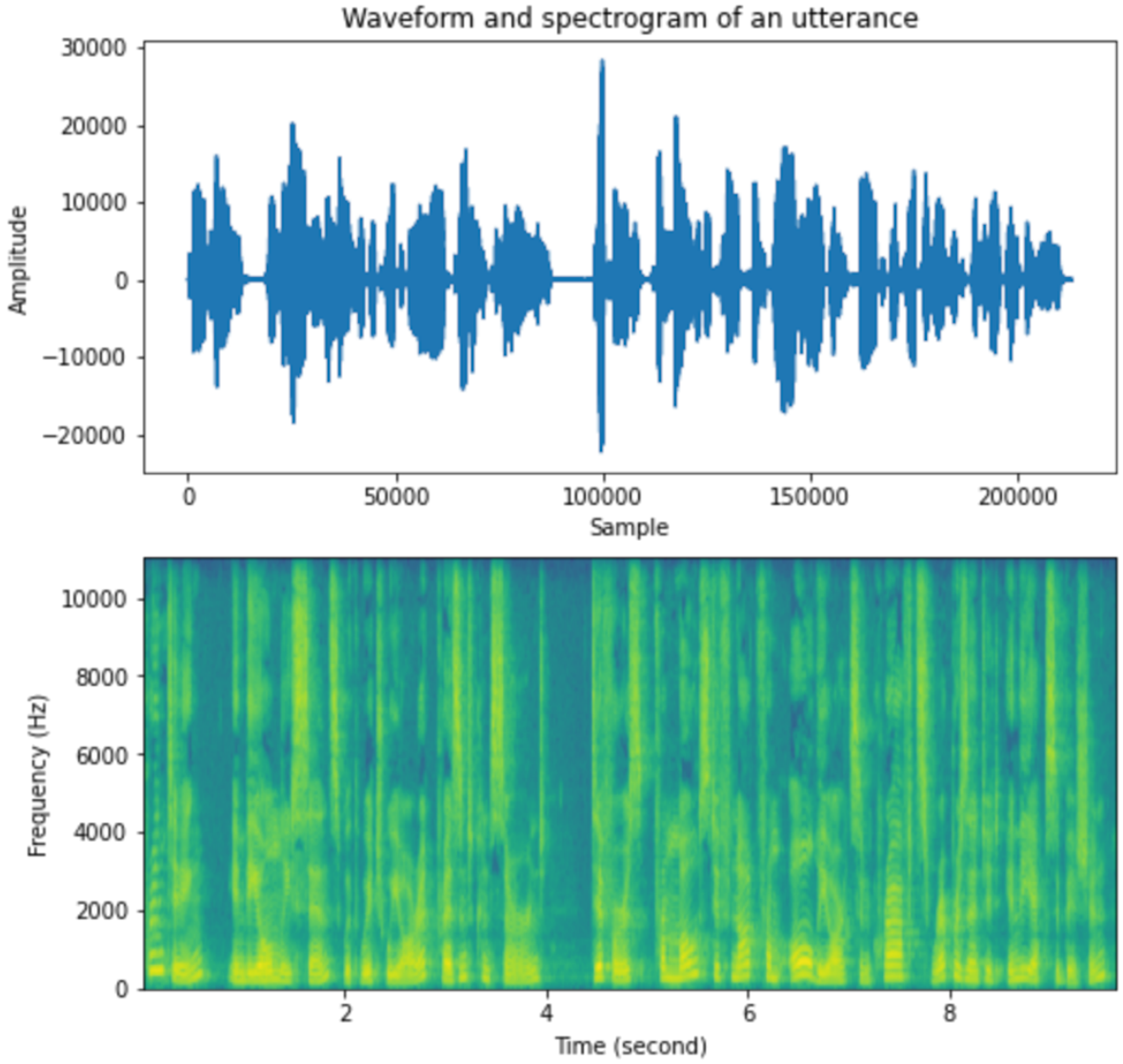

Figure 2.4. An example of a time-domain waveform and its corresponding frequency-domain spectrogram.

Mel-spectrogram

The intermediate representation is mel-spectrogram. It is a compact representation of an audio signal. Tacotron2 is a model that predicts a sequence of mel-spectrum from an input text sequence. WaveNet generates time-domain waveform samples conditioned on the predicted mel-spectrogram.

A spectrogram shows the frequency spectra of a signal over time. To obtain one spectrum, the signal is windowed. The short-time Fourier transform (STFT) is applied to this windowed signal segment to decompose it into its frequency components. By sliding the analysis window over the whole signal with a fixed step size, the spectrogram of the whole signal is constructed by juxtaposing the frequency spectra thus obtained in time. The resulting three-dimensional data could be plotted in a 2D graph. An example of computed mel-spectrogram on one utterance: "Printing, in the only sense with which we are at present concerned, differs from most if not from all the arts and crafts represented in the Exhibition." is shown in Fig. 2.4. The x-axis is time. The y-axis represents the frequency range. The intensity of each point represents the amplitude of the frequency component in the decibel scale. As humans perceive sound frequencies in a non-linear manner, the mel-scale [43] was invented. The scale is designed such that the pitch is perceived linearly increasing along the mel-scale.

Tacotron2

Tacotron2 is an attention-based sequence-to-sequence model. The model architecture is shown in Fig. 2.5. It consists of an encoder, a decoder, and a location-sensitive attention module.

The encoder converts the input text to an internal representation. The input text is a sequence of tokens. The tokens can be the characters, phonemes and punctuation symbols in the support language. A lookup table contains a weight vector for each possible token. The input tokens are transformed to the embedding representation by reading the lookup table. A multi-layer convolution operation is then applied on the embedding representation to capture the localized features among the adjacent tokens. For instance, with the filter length of 5, the convolution layer takes into account the two previous and two next

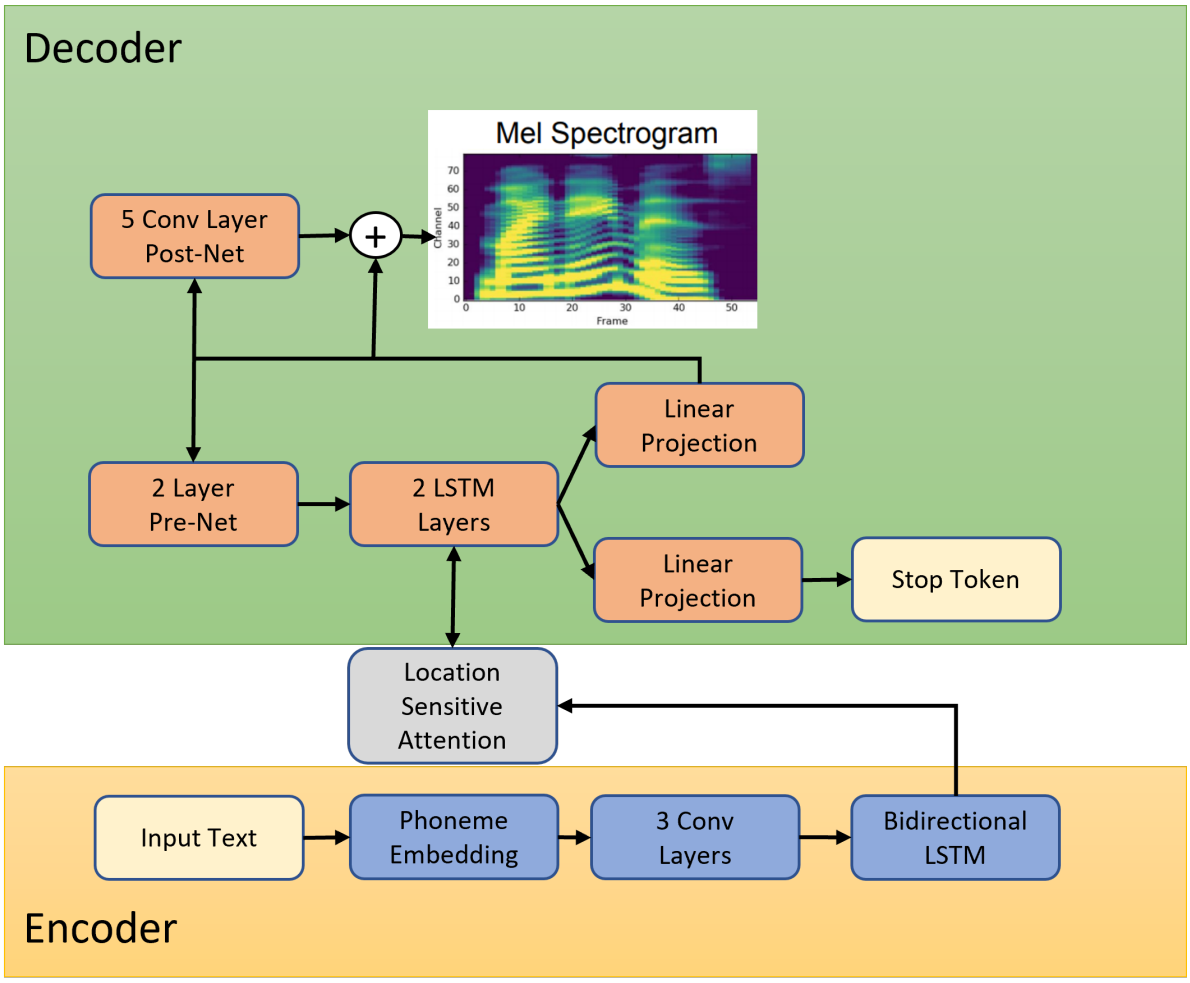

Figure 2.5. Tacotron2 architecture.

tokens in its feature extraction. Each convolution layer is followed by batch normalization and ReLU activation. Finally, a sequence processing unit, bi-directional LSTM, scans the extracted features in both forward and backward directions. The outputs from the two passes are concatenated and become the internal representation.

The decoder converts the internal representation into a mel-spectrogram. Tacotron2 uses the attention mechanism. At each decoding step, a context vector is computed for the decoder. The context vector is a weighted sum of the encoder outputs. The context vector is recalculated for each decoding step, allowing the decoder to focus on the right portion of the encoder outputs during the mel-spectrum generation. Therefore, learning attention is a crucial step for the model. In addition, Tacotron2 is an auto-regressive network. It uses the output of the last decoding step to predict the mel-spectrum in the current step.

The attention mechanism is explained step-by-step here. Assume it is at the time step $\boldsymbol{i}$. The Pre-Net module projects the low dimension mel-spectrum $\boldsymbol{y_{i-1}}$ to a higher dimension vector $\boldsymbol{p_{i-1}}$. The Pre-Net output $\boldsymbol{p_{i-1}}$, and the last context vector $\boldsymbol{c_{i-1}}$ are passed to an Attention LSTM. The Attention LSTM contains a hidden state vector of last time step $\boldsymbol{s_{i-1}}$. It generates the current state vector $\boldsymbol{s_i}$. The current state vector $\boldsymbol{s_i}$, encoder output matrix $\boldsymbol{H}$, last attention weight vector $\boldsymbol{a_{i-1}}$ are used to calculate a new attention weight vector $\boldsymbol{a_i}$ according to equations (2.1) and (2.2). Multiplying the attention weight vector $\boldsymbol{a_i}$ with encoder output matrix $\boldsymbol{H}$ gives the current context vector $\boldsymbol{c_i}$. The current context vector $\boldsymbol{c_i}$ and current state vector $\boldsymbol{s_i}$ are passed to a Decoder LSTM. The Decoder LSTM contains a hidden state vector of the last time step $\boldsymbol{d_{i-1}}$. The output of Decoder LSTM $\boldsymbol{d_i}$ is passed to a linear projection layer to create the mel-spectrum $\boldsymbol{y_i}$ at the current time step. $\boldsymbol{d_i}$ is also passed to another linear projection layer to predict the "stop token", which signals the end of decoding. The initial hidden states of Attention LSTM and Decoder LSTM are set to zeros.

$$\boldsymbol{e_{i,j}} = \boldsymbol{v}^{\mathsf{T}} \tanh(\boldsymbol{W} s_i + \boldsymbol{V} h_j + \boldsymbol{U} a_{i-1} + \boldsymbol{b}) \tag{2.1}$$

$$\boldsymbol{a_i} = \mathrm{Softmax}(e_i) = \frac{\exp(e_i)}{\sum_j \exp(e_j)} \tag{2.2}$$

where i is the decoder time step, j is the $\boldsymbol{j^{th}}$ position of the input text token, and $\boldsymbol{h_j}$ is the corresponding vector in the encoder output $\boldsymbol{H}$. $\boldsymbol{v}$, $\boldsymbol{W}$, $\boldsymbol{V}$, $\boldsymbol{U}$ are the trainable parameters.

The equation in (2.1) makes use of the last attention weight vector $\boldsymbol{a_{i-1}}$ in its calculation of the current attention weight vector $\boldsymbol{a_i}$; hence, the attention mechanism is called "location-sensitive" attention. An alignment matrix of a speech is created by joining the attention weight at each time step. An example is plotted in Fig. 2.6. The x-axis is the decoder timestep. The y-axis is the position index of the input text tokens. At each decoder timestep, only a few of the input tokens receive a high attention score, implying the decoder places its focus only on those tokens.

The Post-Net is a module for improving the spectrogram quality. It is a stack of convolutional layers and batch normalization layers. It computes a residue from the generated mel-spectra. The final mel-spectra is the summation of generated mel-spectra and residue.

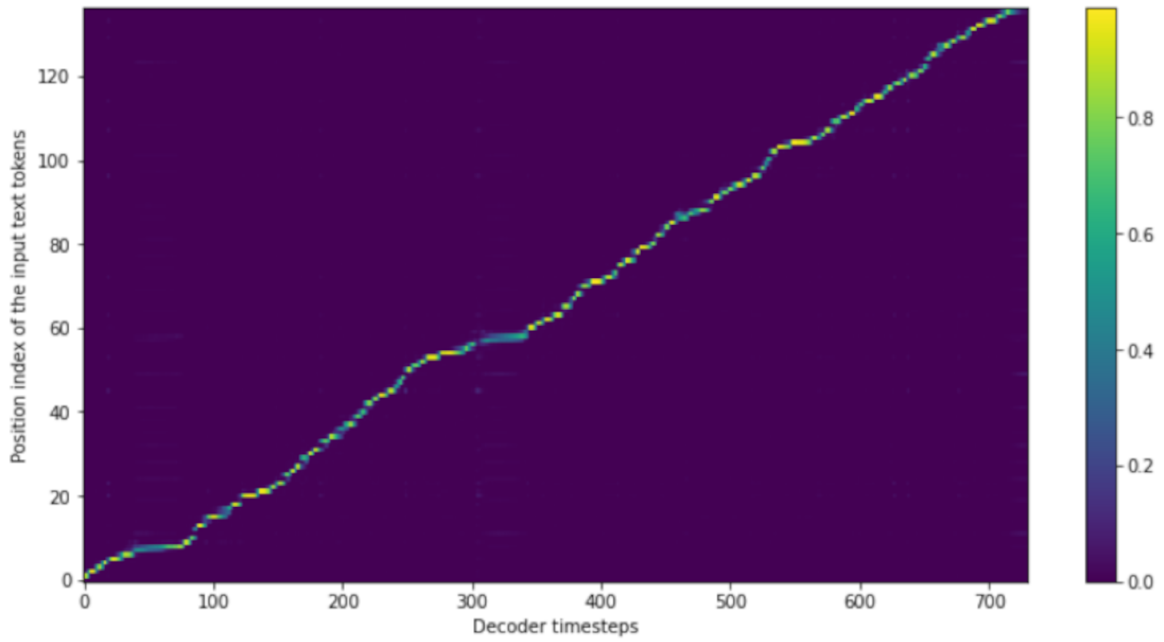

Figure 2.6. An example of visualizing the alignment matrix.

WaveNet

WaveNet is a neural vocoder. In contrast to the classical vocoders, it does not make use of the source-filter model. It is simply a data-driven auto-regressive model. It factorizes a waveform as a product of conditional probability as follows:

$$p(x) = \prod_{t=1}^{T} p(x_t | x_1, ..., x_{t-i}) \tag{2.3}$$

where x_t is the audio sample at time step t.

The model uses the dilated convolutional layers to increase the receptive field of the convolutions. As shown in Fig. 2.7, the receptive field increases exponentially with the number of hidden layers: The hidden layer directly on top of the input has a dilation factor of one, its output depends on only two neighbouring inputs values. The next hidden layer has a dilation factor of two, hence its output is actually dependent on four neighbouring input values, and so on. Therefore, a few layers of stacked convolution gives a very large receptive field to the model. The careful design of the WaveNet architecture enables it to generate high-quality speech matching human's naturalness.

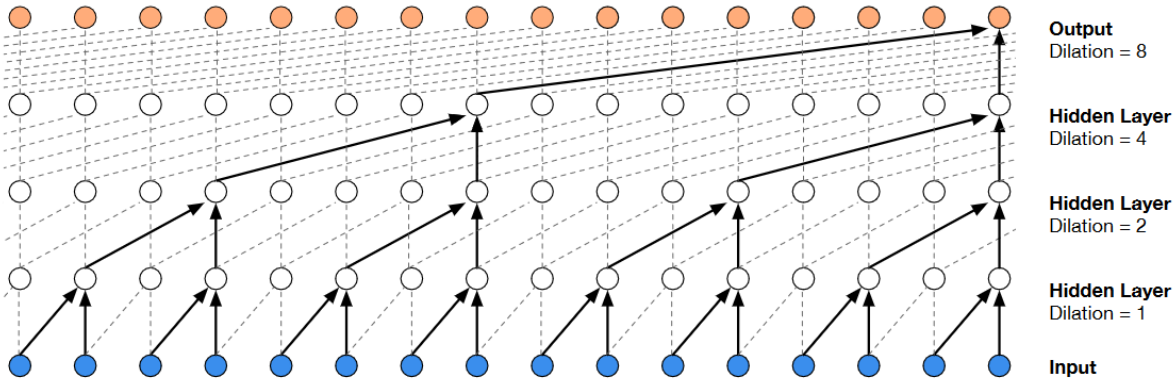

Figure 2.7. Visualization of a stack of dilated causal convolutional layers. The figure is extracted from [48].

The network does not have a recurrent neural module, so the model training is fast. However, during inference, the auto-regression nature implies a sequential audio sample generation. It results in a slow generation speed and is an obstacle for WaveNet's production use. Eventually, some flow-based vocoder models [34, 37] are proposed for real-time speech production without degrading the generated speech naturalness.

2.3 Expressive TTS

The Tacotron2 model is very extensible. To make its output more expressive, it is extended by conditioning on the output of a global style token (GST) layer, which extracts the style information from a reference audio, so that it could generate speech with a similar global style as the reference audio [52].

The GST extractor module contains a reference encoder, and it uses the input audio to predict the style embedding as a weighted combination of the style tokens. The GST module is depicted in Fig. 2.8. The mel-spectra is computed from the input audio. The reference encoder converts the mel-spectra into a style vector. It consists of multiple convolution layers for feature extraction, and a unidirectional GRU to summarize the features over time. The resulting style vector is used as the query vector for multi-head attention. By comparing this style vector over individual style tokens, a weight for each style token is found. The style embedding is the weighted sum of the style tokens.

Wang et al. trained a GST-TTS model with a single storytelling speaker dataset in [52]. According to their analysis, varying the weights of the style tokens could alter the speed and "animated" level of the synthesized speech. The global style tokens learned from just a single speaker is adequate to model the speech prosody in Tacotron2.

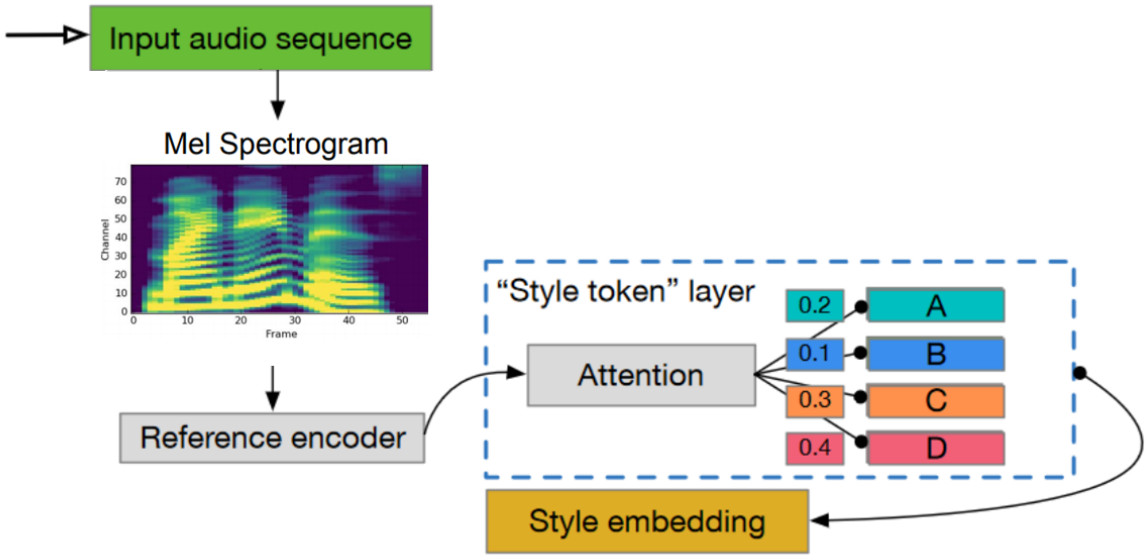

Figure 2.8. The global style token (GST) extractor module.

The style embedding is then concatenated with the output of Tacotron2's encoder. The decoder uses this concatenated result to generate the mel-spectrogram with the same procedure described in Section 2.2.2. The GST Tacotron2 model is depicted in Fig. 2.9.

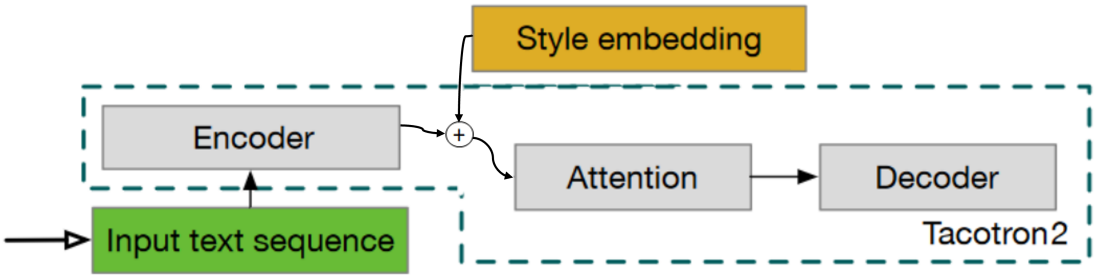

Figure 2.9. GST Tacotron2 architecture.

Stanton et al. [42] demonstrated that the style embedding could be predicted from

texts using a text-predicted GST network. However, they only conducted experiments on storytelling speech data.

2.3.1 Scenario-based TTS

Prateek et al. [36] collected 20 hours of neutral speaking utterances and 4 hours of newscasting utterances from one speaker to train a bi-style text-to-speech model. It used one-hot style embedding and word contextual embedding as additional inputs for its decoder to generate scenario-specific utterance. Hu et al. [12] recorded three speaking styles (neutral, whispered, Lombard) for a speaker on the same texts to create a multi-style text-to-speech model of the speaker. It used a well pre-trained speaker verification model to extract acoustic features from utterances and observed that utterances of the same style clustered together well. The TTS model then based on the mean vector of each style cluster to generate stylish utterances of the speaker.

2.3.2 Emotional TTS

Lee et al. [24] collected 21 speech hours of six different emotions from one female Korean actor. A trainable emotion embedding was added as one more conditional input to Tacotron [51]. Noé et al. [45] investigated the fine-tuning of a pre-trained neutral TTS model with around 20 minutes of emotional speech. This resulted in one model per emotional category. Whitehill et al. [53] proposed an adversarial cycle consistency training method for multi-reference style transfer using partially disjoint datasets. Speaker embeddings and style embeddings were separately trained using the scheme with paired and all possible combinations of unpaired triplets. A triplet consists of synthesizing text, and two reference audios with matched or unmatched styles. The paper shows good performance for an internal dataset consisting of only two speakers, one speaking in neutral style whereas another speaker speaking with neutral style and 3 other emotions.

2.4 Accent Conversion

Deep learning has been applied successfully for accent conversion (AC) in recent years. For example, Liu et al. [28] firstly pre-trained a speaker encoder from multiple corpora (LibriSpeech, VoxCeleb1, and VoxCeleb2) consisting of around 8.4k mixed-accent speakers. The encoder was used to extract speaker embedding from an utterance. A TTS model with this pre-trained speaker encoder was trained with multi-speaker native speeches. Consequently, the TTS model could synthesize speech with only a native accent reasonably well for a new speaker. To obtain the linguistic representation from speech, the model had an automatic speech recognition (ASR) model that was pre-trained with native speeches. The ASR model was then adapted to a non-native speaker with a known accent label. The training objective of the ASR model was to take in an L2 speech and produce a linguistic representation that is expected from a native speaker speaking the same utterance. To achieve it, their model was trained with the VCTK corpus [55], which contains 110 speakers of various accents uttering the same set of sentences.

Li et al. [26] proposed another accent conversion approach. Their model converted the speech of a native speaker to a non-native speaker's voice. The model is shown in Fig. 2.10. It had a reference encoder to extract prosodic information from the native speech. Like [28], it had an ASR model, trained on 10,000 hours of mixed accent data, to extract the linguistic representation from the speech. The AC model was based on Tacotron2 [40], and the synthesizer was trained with utterances from the L2 speaker. During model inference, the synthesizer utilized speaker-independent linguistic representation and prosodic information extracted from the native utterance to generate a speech with the L2 speaker's voice in a native style. However, as the synthesizer was trained with only L2 utterances, it may generate speech with some incorrect pronunciations and phonemes. We compared their audio samples against ours in the experimental section. The authors of [28, 26] also reported the loss of voice similarity in their synthesized speech.

Although an ASR model could extract linguistic representation from speeches, if an L2 speech is wrongly recognized, the synthesized accent-reduced speech would become less intelligible due to pronunciation errors. Recent state-of-the-art general-purpose ASR engines developed by the big research arms from Microsoft, Google, Amazon and the

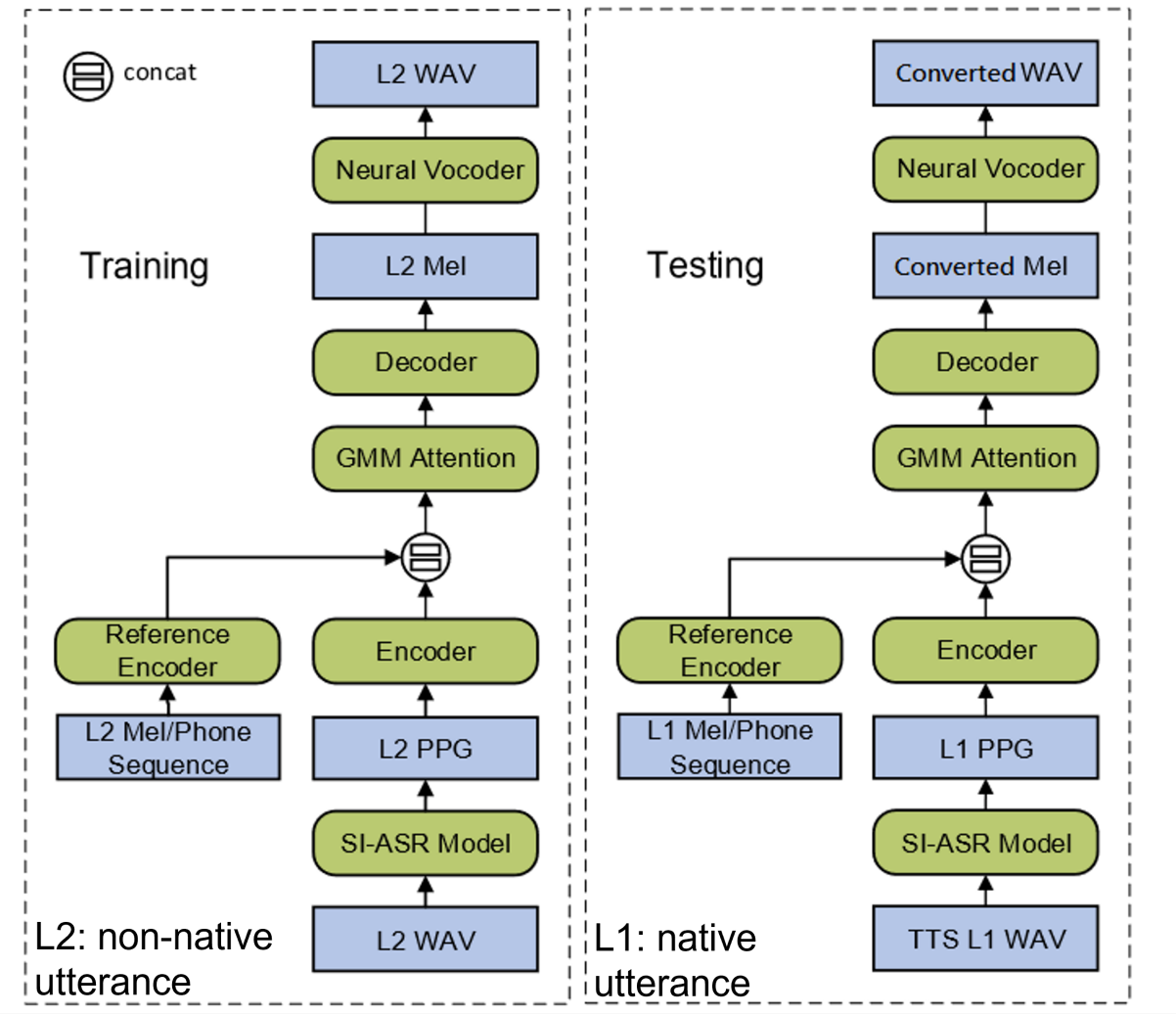

Figure 2.10. Accent conversion model proposed in [26].

like, and ASR models designed for accented speech [20] can obtain a highly accurate transcription of non-native utterances. Hence, our current work focuses on text-to-speech for an L2 speaker with text inputs, rather than voice conversion. The goal is to produce L2 synthesized speech with the voice of the non-native speaker in a more native accent and prosody.

2.5 Data Augmentation in TTS

Data augmentation is very common in deep learning. In image classification, AlexNet [11] applied on-the-fly data augmentation by translating and flipping training image data.

Zhu et al. [65] generated some virtual training samples for improving the performance of a speaker verification model.

Few works used data augmentation in text-to-speech model training. Lee et al. [27] first pre-trained a TTS model and an ASR model, and then applied the ASR model to improve the clarity of augmented speeches generated from unpaired combinations of GSTs and text contents. Dipjyoti et al. [35] applied a classic signal processing technique, Spectral Shaping and Dynamic Range Compression (SSDRC), to convert 2 hours of normal speech of a speaker to Lombard-style speech. Then they fine-tuned a Tacotron model, which was pre-trained with a single speaker's speech corpus, LJSpeech [16], with 0.5 hours of real Lombard speech uttered by the speaker with the additional two hours of converted speech. Huybrechts et al. [14] proposed a 3-steps approach on leveraging some augmented data. They trained a voice conversion (VC) model, CopyCat [18], to convert available stylish speech to the target speaker's voice. Then they trained their text-to-speech model with these additional augmented data. Because of the quality issue of some converted audios, they had to finetune their model with a real stylish speech from the target speaker to obtain synthesized stylish speech with better quality. Zhao et al. [60] extended previous accent conversion works by training a pronunciation correction model to perform accent reduction in an utterance using training pairs of foreign-accented speech and synthesized accent-reduced speech, both uttered by the same speaker. The pronunciation correction model is a voice conversion model.

CHAPTER 3

PROPOSED METHOD

In this chapter, we first introduce our TTS model for speech imitation. Then we will present our proposed data augmentation method, and put forward two TTS model designs for the target applications.

3.1 TTS Model for Speech Imitation

Our model is based on Tacotron2 [40]. Details of which are given in section 2.2.2. It also consists of the GST module described in section 2.3.

3.1.1 Speaker Embedding and Speaking Style Embedding

Since our dataset contains multiple speakers, we use speaking embeddings to model the speakers. Each speaker is represented by a speaker embedding vector of 128 dimensions. The speaker embedding vector is randomly initialized and then jointly trained with Tacotron2.

We use the same GST module design as in [52] to compute the speaking style embedding S_{gst}. The ground-truth GST extracted from a reference audio during training is denoted as S_{gt}:

$$S_{gt} = \mathrm{GST}(audio)\,. \tag{3.1}$$

The speaker embedding and speaking style embedding are channel-wise concatenated with Tacotron2's encoder outputs. The ground-truth mel-spectrogram of an input audio is denoted as MEL and Tacotron2 model is denoted as Taco.

$$\mathrm{MEL} = \mathrm{Taco}(text, speaker, S_{gt})\,, \tag{3.2}$$

where Taco consists of an encoder $\mathrm{Taco}^{\mathrm{enc}}$ and a decoder $\mathrm{Taco}^{\mathrm{dec}}$. The loss function of Tacotron2 model is defined as:

$$\mathrm{Loss}_{\mathrm{taco}} = |\mathrm{MEL} - \hat{\mathrm{MEL}}|_2 + \mathrm{BCE}(\mathrm{Stop}, \hat{\mathrm{Stop}}) \,, \tag{3.3}$$

where $\hat{\mathrm{MEL}}$ is the model's predicted mel-spectrogram; Stop is the binary "stop token" which signals the end of decoding and $\hat{\mathrm{Stop}}$ is its predicted value. Based on the length of the target audio, each output frame is assigned 0 or 1 to represent whether it is the stopping frame. A binary classifier uses the output of the decoder to predict the class label of the stopping frame. The loss function for training this classifier is binary cross-entropy BCE.

Our Tacotron2 learns to perform speech synthesis conditioning on speaker embedding and speaking style embedding. The process is illustrated in Fig. 3.1.

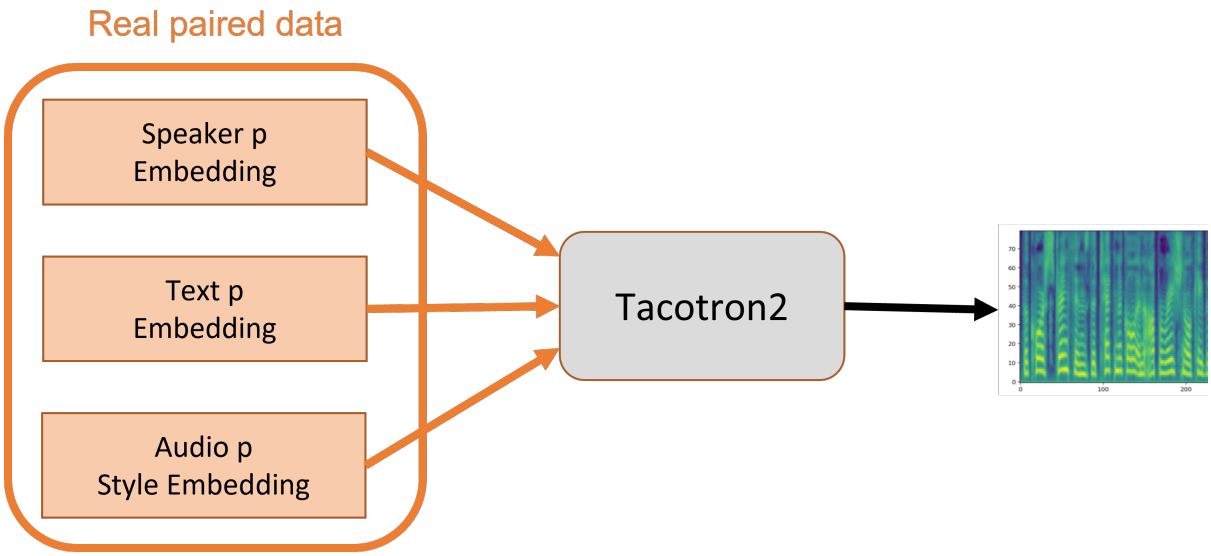

Figure 3.1. Speech synthesis using Tacotron2 by conditioning on speaker embedding and speaking style embedding.

3.1.2 Text-predicted Scenario-based Global Style Tokens

As there will be no ground-truth reference audio during inference, we follow the same idea of [42] and add a text-predicted global style token network (TP-GST) to get the predicted speaking style embedding S_{tp}. We modify its input by concatenating a one-hot style label with the text embedding to compute S_{tp}:

$$S_{\mathbf{tp}} = \mathrm{TPGST}(text, style\ label)\ . \tag{3.4}$$

The modified TP-GST is shown in Fig. 3.2. During model training, we require the text-predicted GST as close to the ground-truth GST extracted from the corresponding reference audio as possible.

$$\mathrm{Loss}_{\mathrm{tpgst}} = |S_{\mathbf{gt}} - S_{\mathbf{tp}}|\ . \tag{3.5}$$

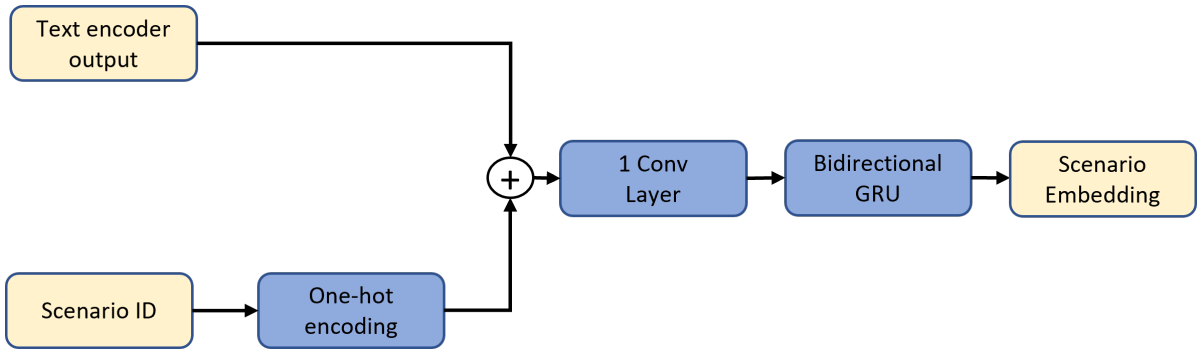

Figure 3.2. The module that computes the text-predicted scenario-based GST.

3.1.3 On-the-fly Data Augmentation for Style Transfer

In the original Tacotron2 training process, an attention mechanism learns an alignment matrix for the alignment relationship between the text input and the generated audio frames. We use $\boldsymbol{h}_{\mathbf{text}}$ to represent the hidden state sequence of an input text, $\boldsymbol{h}_{\mathbf{p}}$ to represent the trainable embedding of a speaker p, $\boldsymbol{S}_{\mathbf{p}}$ to represent the style embedding of a speech from speaker p, $\boldsymbol{A}_p$ to represent the corresponding alignment matrix produced by Tacotron2. During Tacotron2 encoding step, its encoder generates the hidden state sequence, the style embedding as well as the speaker embedding for the given inputs as follows:

$$\boldsymbol{h}_{\mathbf{text}}, \boldsymbol{S}_{\mathbf{p}}, \boldsymbol{h}_{\mathbf{p}} = \mathrm{Taco}^{\mathrm{enc}}(speaker, audio, text)\ . \tag{3.6}$$

During Tacotron2 decoding step, its decoder takes the outputs from its encoder, and predicts the output mel-spectrograms using autoregression with the attention mechanism:

$$\boldsymbol{MEL}_p, \boldsymbol{A}_p = \mathrm{Taco}^{\mathrm{dec}}(\boldsymbol{h}_{\mathbf{text}}, \boldsymbol{S}_{\mathbf{p}}, \boldsymbol{h}_{\mathbf{p}})\ . \tag{3.7}$$

A by-product of decoding is the attention alignment matrix, which is usually discarded afterwards in standard Tacotron2 training. We argue that the alignment matrix actually encapsulates useful rhythmic information of the input text/audio. Observing that speeches under different scenarios show different rhythm characteristics, we design our model to strengthen the influence of the speaking style embedding on the alignment matrix. To achieve this goal, we generate augmented data for any speaker q by having him/her imitate the alignment matrix $\boldsymbol{A_p}$ of another speaker p given the text and audio from the latter speaking in style $\boldsymbol{S_p}$ which speaker q does not speak with. That is, if we denote the mel-spectrogram of the augmented data from speaker q as $\boldsymbol{MEL_q}$, and the corresponding alignment matrix as $\boldsymbol{A_q}$, we will have

$$\boldsymbol{MEL_q}, \boldsymbol{A_q} = \mathrm{Tacotron}^{\mathrm{dec}}(\boldsymbol{h_{\mathrm{text}}}, \boldsymbol{S_{\mathrm{p}}}, \boldsymbol{h_{\mathrm{q}}}) \,, \tag{3.8}$$

where q is not equal to p.

As the augmented data may not be perfect, we focus only on its style property captured by the alignment matrices, and incorporate the alignment loss only during the training of augmented data (for any unpaired speaker and style combinations), which is defined as the Frobenius norm of the difference between the two alignment matrices as follows:

$$\mathrm{Loss}_{\mathrm{align}} = |\boldsymbol{A_p} - \boldsymbol{A_q}|_2 \,. \tag{3.9}$$

Note that the alignment loss is not applied in the training of the real paired data, it only applies to the augmented unpaired data to sharpen the desired rhythm characteristics.

A similar alignment loss was also utilized by [64] in their guided attention method with pre-aligned phoneme sequences to speed up and stabilize their Tacotron2 training.

Note that this data augmentation is performed for any two different speakers speaking in different styles on all training utterances from all speakers from disjoint datasets in this thesis. This data augmentation scheme is further illustrated in Fig. 3.3. In this figure, the real paired data contain the speaker embedding of speaker p, together with text and GST extracted from his/her training utterance. The augmented unpaired data is derived from the same text and GST but using another speaker embedding from, say, speaker q. Consequently, in a mini-batch update, the model will be trained with the real sample pairs and augmented sample pairs.

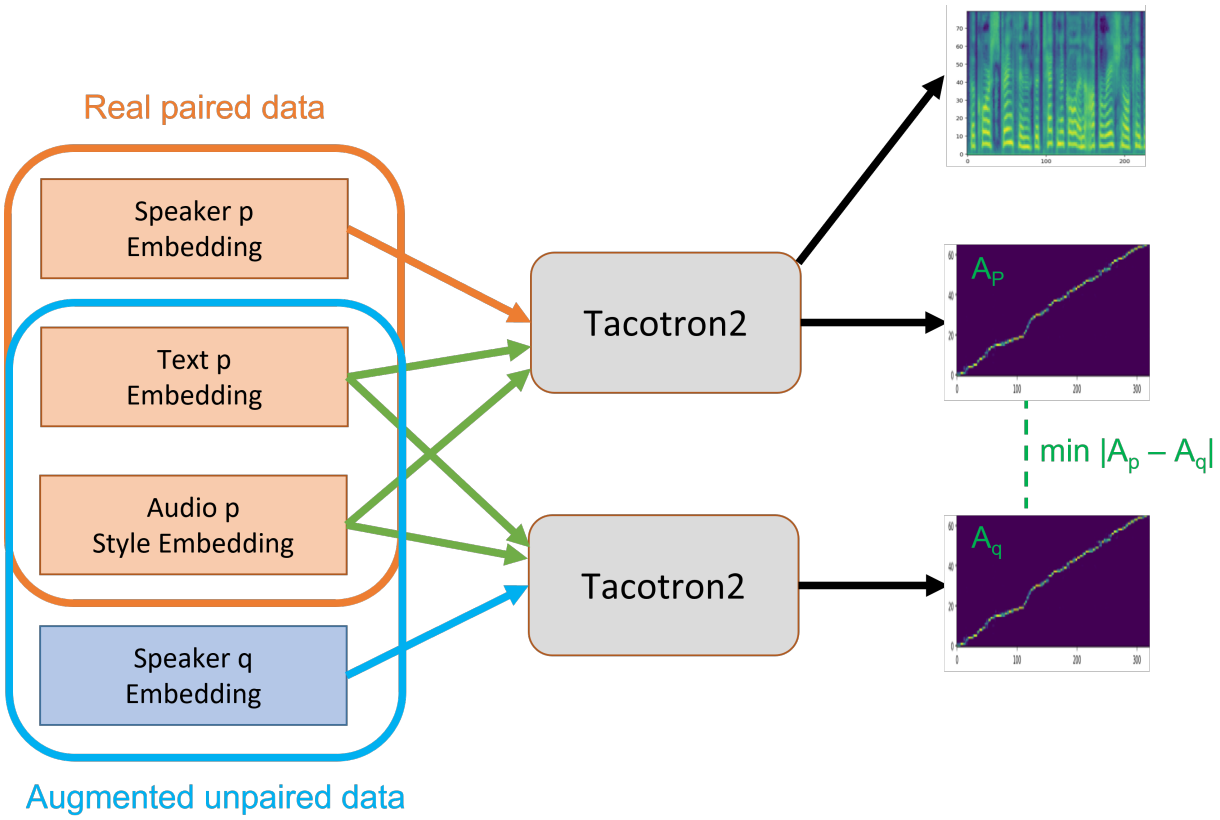

Figure 3.3. Proposed on-the-fly data augmentation scheme.

There are two choices for the data augmentation scheme: (1) *many-to-one data augmentation only for the neutral speaker*: generate stylish augmented data only from originally non-neutral stylish speeches with speaker q being the neutral speaker; (2) *many-to-many data augmentation for all speakers*: generate an augmented sample for any original speech spoken by speaker p for each speaker $q \neq p$, resulting in multiple additional augmented samples for each original audio.

Finally, the overall loss function for training our proposed model is the sum of the three losses in predicting the output mel-spectrograms, GSTs and alignments:

$$\text{Loss} = \text{Loss}_{\text{taco}} + \text{Loss}_{\text{tpgst}} + \text{Loss}_{\text{align}} \,. \tag{3.10}$$

3.2 Scenario-based TTS model

The global speaking styles of speech spoken in different scenarios have perceptible differences in acoustic expressiveness, such as speech tempo and pitch variation. The proposed

scenario-based TTS model is designed to transfer the global speaking style from stylish speakers to one neutral-tone speaker.

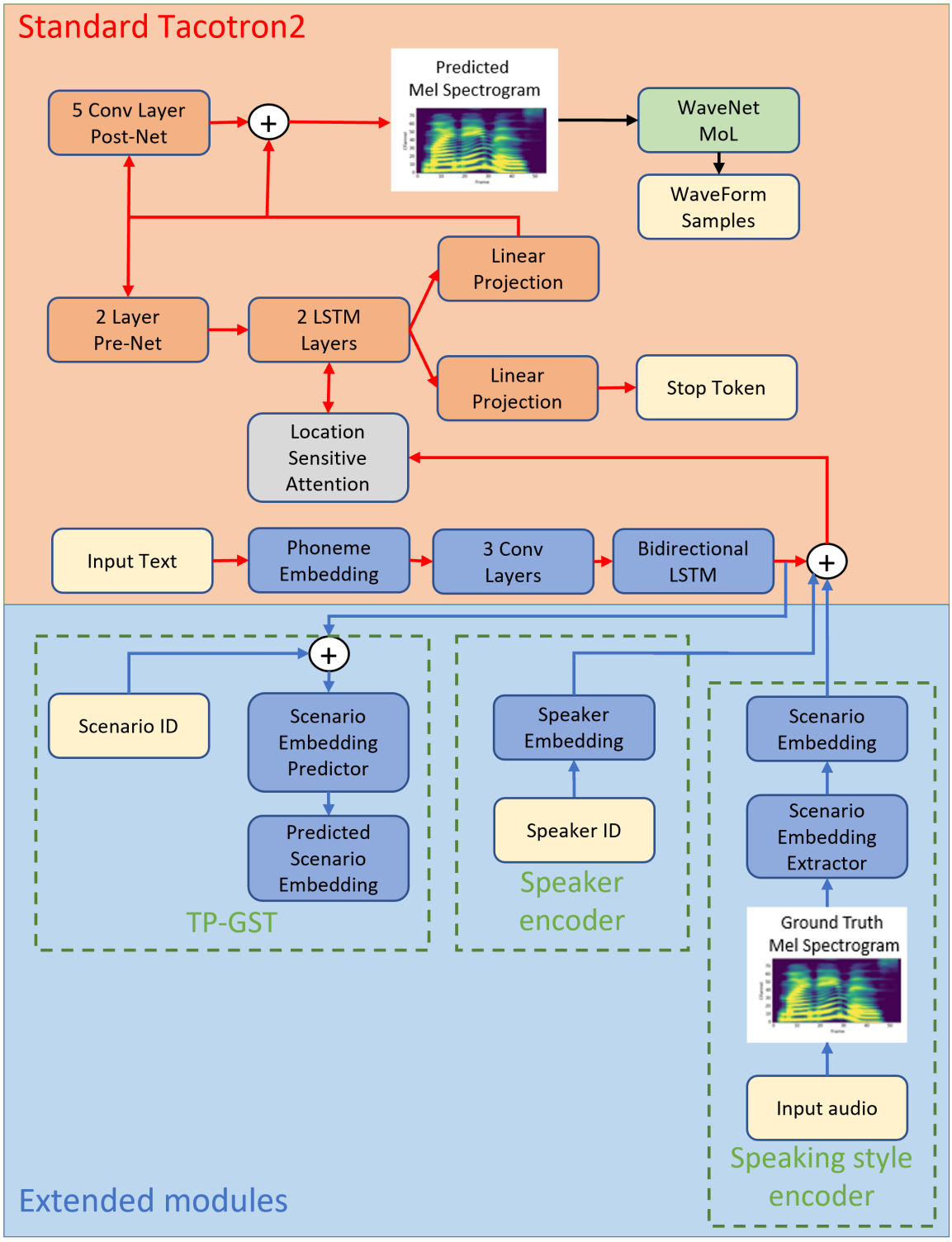

Figure 3.4. Proposed scenario-based Tacotron2 model under training mode.

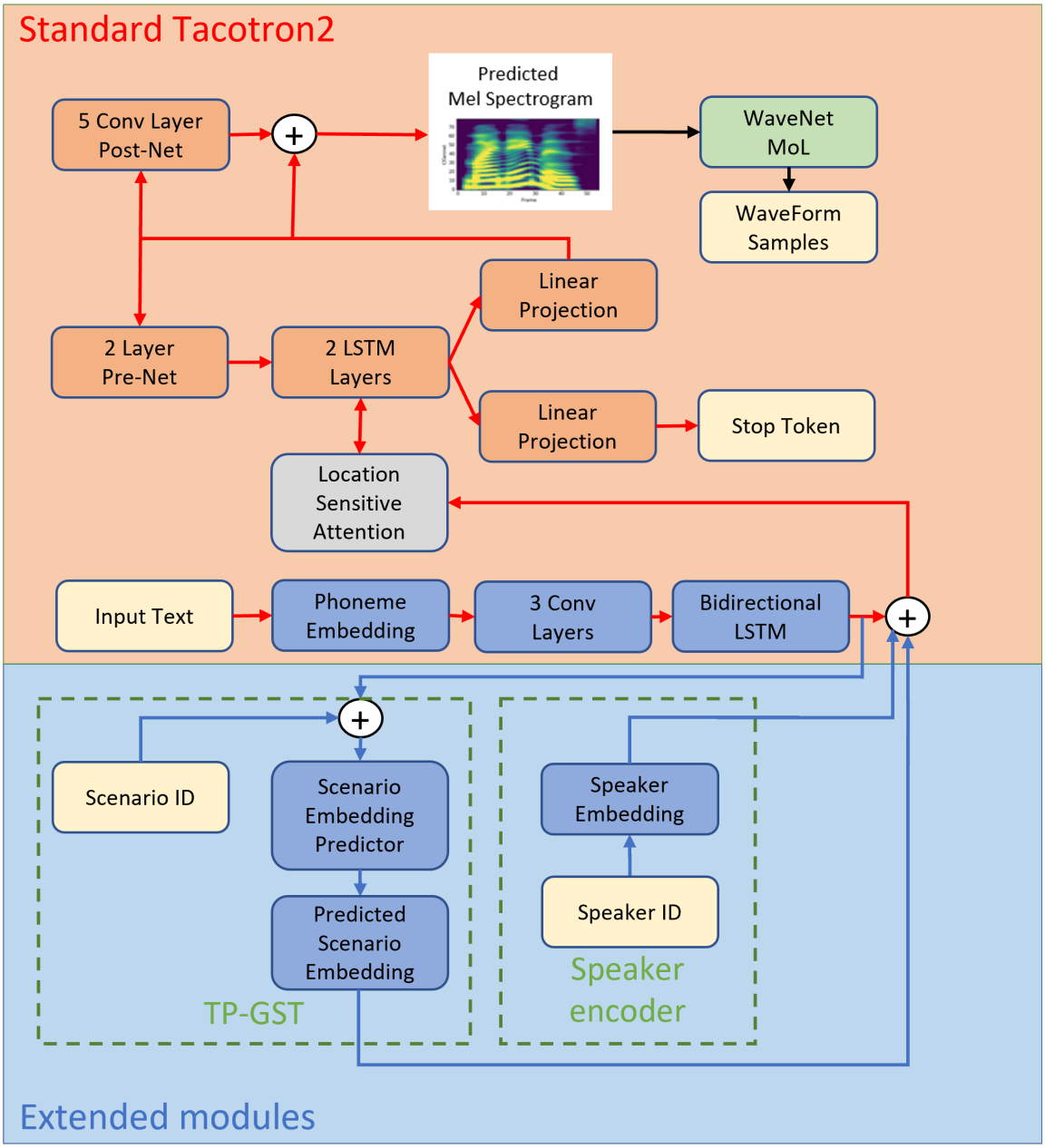

Figure 3.5. Proposed scenario-based Tacotron2 model under inference mode.

The model under the training mode and inference mode are illustrated in Fig. 3.4 and Fig. 3.5 respectively. The standard Tacotron2 consists of the components connected with red arrows. The components connected with blue arrows are the proposed extended modules. During training, the speaker encoder provides the speaker embedding. The speaking style encoder is an implementation of the GST module. Under this context, its input is the different scenario speeches. Hence, the extracted speaking style embedding is referred

to as scenario embedding. The TP-GST module is trained to use the text embedding and scenario ID to predict the scenario embedding extracted from the input audio. During inference, there will be no input audio, and the speaking style encoder is not usable. TP-GST supersedes the speaking style encoder to provide a predicted scenario embedding for speech synthesis of Tacotron2.

3.3 Accent-beautified TTS model

The common perceptible differences of non-native speech from native speech are the wrong prosody (e.g., rhythm, pause occurrence, word duration, and intonation pattern) and word mispronunciation. These two factors contribute to foreign accentedness. The proposed accent-beautified TTS model is designed to reduce the foreign accentedness of a non-native speaker.

The model under the training mode and inference mode are illustrated in Fig. 3.6 and Fig. 3.7 respectively. The standard Tacotron2 consists of the components connected with red arrows. The components connected with blue arrows are the proposed extended modules. During training, the speaker encoder provides the speaker embedding. The speaking style encoder is an implementation of the GST module. Under this context, its input is either native or non-native speech. Hence, the extracted speaking style embedding is referred to as accent embedding. The TP-GST module is trained to use the text embedding and native or non-native label to predict the accent embedding extracted from the input audio. During inference, there will be no input audio, and the speaking style encoder is not usable. TP-GST supersedes the speaking style encoder to provide a predicted accent embedding for speech synthesis of Tacotron2. The global speaking style from native accent weighted GSTs reduces the wrong prosody of non-native speech.

In addition, we propose to include a phoneme recognition module during training to "force" the TTS model to generate more "correct" mel-spectrograms from the ASR perspective. The phoneme recognition module is pre-trained with a native speaker's speech, and it is applied on the mel-spectrogram generated by Tacotron2's decoder. During inference, this phoneme recognition module is not used. This additional module during model training reduces the pronunciation errors of the synthesized speech from a non-native speaker.

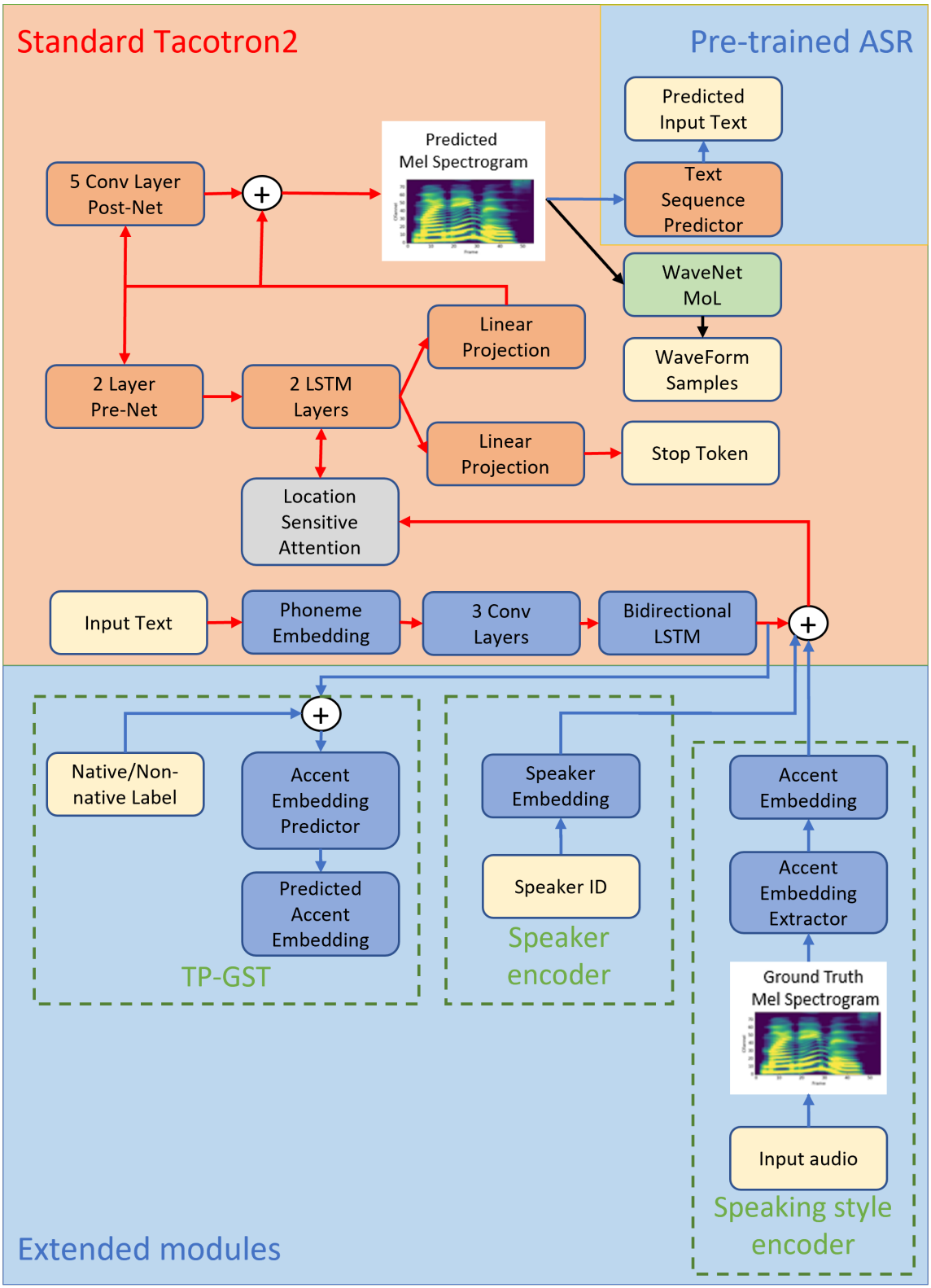

Figure 3.6. Proposed accent-beautified Tacotron2 model under training mode.

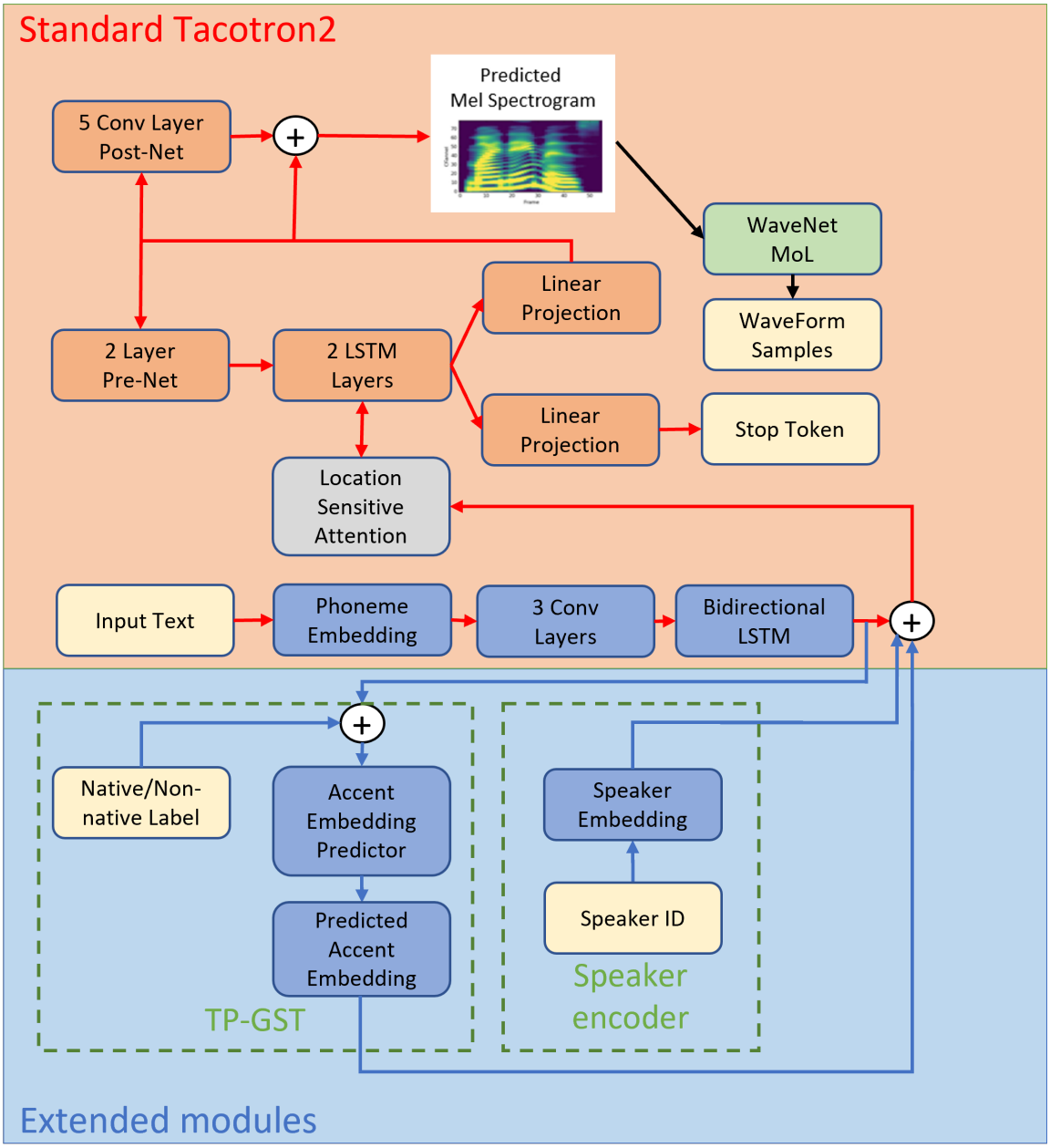

Figure 3.7. Proposed accent-beautified Tacotron2 model under inference mode.

3.3.1 Phoneme Recognition Module

A combination of a TTS model with an ASR module was proposed in [27] to enhance the clarity of the synthesized speech with a pre-trained Listen, Attend and Spell (LAS) model [5]; we use a simplified version of the Deep Speech model [9] instead with mel-spectrogram as input. The module is illustrated in Fig. 3.8. It consists of layers of linear

projections and a Bi-RNN layer, and a softmax classifier is used to predict the phoneme sequence of the input. Thus, we have

$$\hat{X_{seq}} = \mathrm{ASR}(\mathrm{MEL}) \,, \tag{3.11}$$

where MEL is the mel-spectrogram and $\hat{X_{seq}}$ is the predicted phoneme sequence. Connectionist temporal classification (CTC) [8] is used to quantify the mismatch between the target sequence and the predicted sequence. No language model is used to rectify the predicted sequence. We define the ASR loss as follows:

$$\mathrm{Loss}_{\mathrm{asr}} = \mathrm{CTC}(\hat{X_{seq}}, X_{seq}) \,, \tag{3.12}$$

where X_{seq} is the target phoneme sequence.

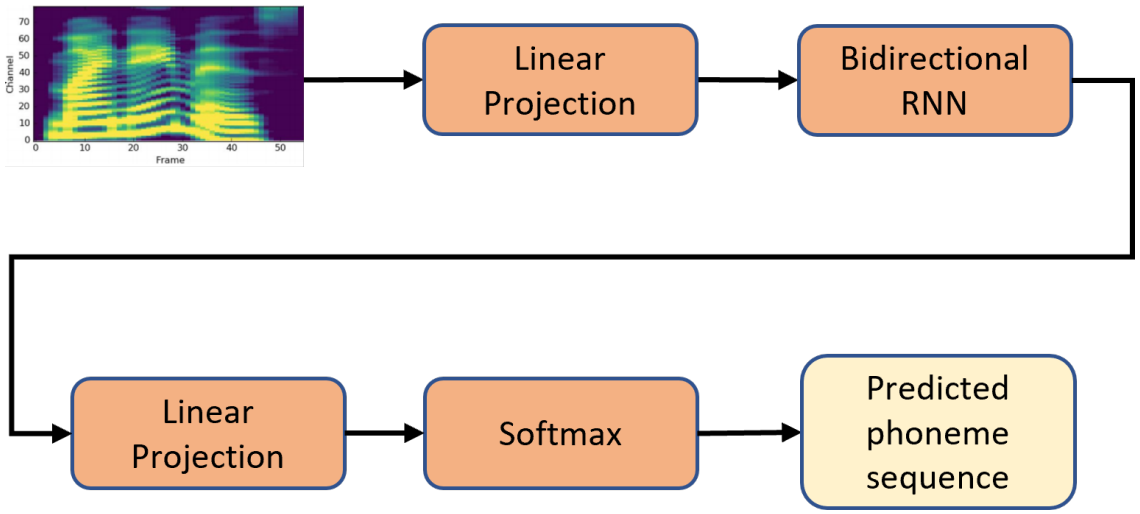

Figure 3.8. The phoneme recognition module.

Finally, the overall loss function for training our proposed model is the sum of losses in predicting the target mel-spectrograms, GSTs, and alignments together with the additional ASR loss:

$$\mathrm{Loss} = \mathrm{Loss}_{\mathrm{taco}} + \mathrm{Loss}_{\mathrm{tpgst}} + \mathrm{Loss}_{\mathrm{align}} + \mathrm{Loss}_{\mathrm{asr}} \,. \tag{3.13}$$

We define the first two terms as TTS loss, the third term as DA loss, and the last term as ASR loss. In the proposed model design, the ASR loss is used in the accent-beautified TTS model because the training data contains non-native accent speeches. Our experimental

result shows that including such term could further improve the clarity of the non-native speech. In the scenario-based TTS, as all speakers speak in their native language, ASR loss is not needed.

Last but not least, our proposed method has a key advantage in comparison to the aforementioned related works. It does not require any speaker to record utterances in a speaking style of more than one scenario, nor does it require the speakers to utter the same set of sentences.

CHAPTER 4

EXPERIMENTAL EVALUATION

In this chapter, we present the data collection and pre-processing pipeline. Then we describe our experiments and provide the result analysis.

4.1 Dataset Collection and Pre-processing

We used both academic research data and data collected from the web. Generally, utterances in academic research datasets are recorded in a controlled environment, typically in a silent room. Hence, they are of higher audio quality. Unfortunately, we did not find suitable training data for some target scenarios. Furthermore, we are interested to know whether the scenario-specific audio data readily found from the internet can be used for multi-style text-to-speech model training. So we mined data from the internet.

We explored different audio sources from the web, including audiobooks from Google Play Store, news reporting websites, public speaking videos on YouTube. Firstly, we downloaded the data and converted them to the same audio format of 16 kHz sampling rate. As we needed high-quality audio for text-to-speech model training, we reviewed the audios to exclude the segments that were not good, for instance, audio segments with background music. Then we applied an online speech-to-text engine, Microsoft Azure[1], to obtain the orthographic transcriptions of the collected audios. The next step was to segment the audio data into suitable lengths for model training. The Montreal Forced Aligner [30] was further used to obtain the word alignment between each audio and its transcript. An example is illustrated in Fig. 4.1. The top panel shows the audio waveform. The second panel shows the aligned transcription. The third one shows the phonetic alignment. The alignment information was then used to segment the audio into short clips of 2 to 10 seconds, which is the usual length of a spoken sentence.

[1]https://azure.microsoft.com/en-us/services/cognitive-services/speech-to-text/

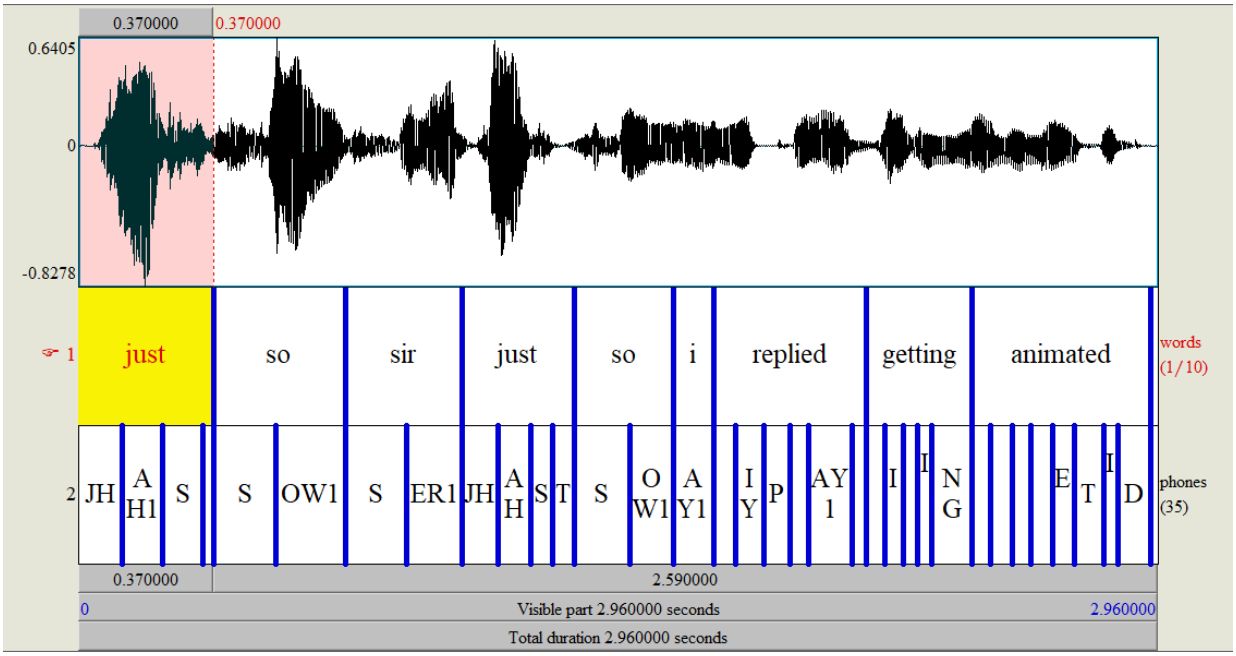

Figure 4.1. An example of audio-to-text alignment by Montreal Forced Aligner.

4.2 Experimental Setup in Common

We modified the codebase[2] of Mellotron [47] to implement our proposed method. The mel-spectra were computed with a hop length of 12.5ms and a window size of 50ms. We used a min-batch size of 64 and the Adam optimizer [21] with a learning rate of 5×10^{-4} during training.

Rather than using WaveNet, we used WaveGlow [37] vocoder. WaveGlow is a flow-based generative network that converts a mel-spectrogram into a waveform. Its design was inspired by Glow [22] and WaveNet [48]. It could generate high-fidelity speech in real-time. We used the version[3] released by Nvidia. Their subjective assessment regarding audio quality showed it reached a similar pleasantness score as WaveNet.

We used Amazon Mechanical Turk (MTurk)[4] to recruit listeners for the subjective test. MTurk is a crowdsourcing platform and is commonly used in similar TTS research works.

[2]https://github.com/NVIDIA/mellotron

[3]https://github.com/NVIDIA/waveglow

[4]https://www.mturk.com/

4.3 Evaluation Metrics

Mean Opinion Score

A Mean Opinion Score (MOS) is a numerical measure of the overall quality of an event or experience by a human. As it is hard to define an objective quality score in line with the perceptive goodness of various aspects of a synthesized utterance such as naturalness, intelligibility, etc, MOS has become a popular approach as a subjective evaluation in a speech quality test. In an MOS evaluation, each participant gives a rate against a sample. The absolute category ranking scale is commonly adopted. The range is shown in Table 4.1. Then the average score rated by all participants is reported.

MOS	1	2	3	4	5
Quality	Bad	Poor	Fair	Good	Excellent

Table 4.1. Five-grade Mean Opinion Score (MOS) scale.

Word Error Rate

The Word Error Rate (WER) is a measure to evaluate the performance of an automatic speech recognition system. Given the original reference text and the transcribed text, the Levenshtein distance is applied to find the distance between the two texts. The distance depends on three types of errors: (a) a substitution error refers to a wrong replacement of a word. For instance, "happy" is transcribed to "floppy"; (b) an insertion error refers to an extra word in the transcribed text; (c) a deletion error refers to a missing word from the reference text.

The format definition of WER is:

$$WER = \frac{S + D + I}{N} \times 100\% \tag{4.1}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, and N is the number of words in the reference text.

4.4 Scenario-based TTS

4.4.1 Dataset

We did not find a voice talent to record speech data of various expressive scenarios. Instead, we collected existing audio data from audiobooks or from the Web which were spoken under four realistic scenarios: neutral speaking, newscasting, public speaking and storytelling. Specifically, to get a large number of neutral speaking audios, we purchased an audiobook from Google Play read by Michelle Obama. Her speaking style was considered neutral in reading the audiobook. We then collected some newscasting audios of a news anchorwoman from the Voice of America (VOA) website. Some Hillary Clinton's public speeches and presidential debate videos on YouTube were collected for the public speaking style. Finally, the children's audiobooks from the Blizzard Challenge[5] 2017 were used for the storytelling style.

The final amounts of collected data of the four wanted styles are summarized in Table 4.2. We held out 20 utterances from each speaker to form the development set. For testing, we held out 50 utterances from Michelle Obama's data, and 30 utterances from each of the other three stylish speakers' data.

Speaker	Scenario	Audio data (hr)
Michelle Obama	Neutral	11.7
VOA	Newscasting	2.1
Hillary Clinton	Public speaking	2.5
Blizzard 2017	Storytelling	3.5

Table 4.2. Summary of collected audios in four speaking scenarios.

4.4.2 Experimental Setup

We started with a pre-trained model checkpoint based on a subset of the LibriTTS dataset [56]. The subset of the LibriTTS dataset contained 123 speakers. Each speaker contributed

[5] A competition organized by the Language Technologies Institute of Carnegie Mellon University on building speech synthesizers on the same set of data.

around 20 minutes of speech data. All of the utterances were less than 10 seconds. We trained our proposed model with the prepared training data in Table 4.2. Since each stylish speaker had less audio data than the neutral speaker's, the audio data of each stylish speaker were repeated in each epoch to around the same amount (around 12 hours) as the neutral speaker's to tackle the data imbalance issue. We trained the model for 80 epochs. Afterward, we continued the model training with the original training data together with on-the-fly augmented data for another 120 epochs. Some synthesized utterance samples could be found on this webpage[6].

4.4.3 Baseline Model

For evaluation, we compared the synthesized utterances from our proposed model with those from a baseline model which was trained with utterances only from the neutral speaker, Michelle Obama. The baseline model is equivalent to a single-speaker text-predicted GST Tacotron2.

4.4.4 Objective Evaluation

The perception of a speech to humans is greatly influenced by two factors: the pitch changing profile and rhythm of speech. One may manipulate these two factors in a TTS system to generate speech of different styles. For instance, we notice the following in our targeted scenarios: In the newscasting scenario, utterances are usually spoken a little faster than neutral speech [36]; in the public speaking scenario, keywords may be stressed, emphasized or lengthened; in the storytelling scenario, speeches are more rhythmic. Therefore, we evaluated the effectiveness of the proposed on-the-fly data augmentation on its effects on the rhythm and pitch profile of the generated stylish utterances. Here, data augmentation was performed on all speakers (2^{nd} choice in Section 3.1.3).

Effect on the Rhythm of Style Transfer to Newscasting Style

We roughly measured the rhythm of a speech by the speaking rate which is defined as the number of phonemes spoken per second. Utterances on the held-out unseen news texts

[6]https://raymond00000.github.io/ttsdemo.html

were synthesized for each speaker, other than the VOA anchorwoman. The newscasting-style utterances synthesized by the TTS model trained with the proposed data augmentation method, were compared with the speakers' speaking rate in their original-style utterances. The results are shown in Table 4.3. Firstly, we notice that the speaking rate of the newscaster is faster than all the other speakers. Secondly, our model successfully synthesizes newscasting utterances for other speakers with a speaking rate close to the newscaster's which is greater than the original speaking rate of the individual speakers. We only performed the evaluation with newscasting as the target scenario style because, from Table 4.3, we notice that the speaking rate of the newscasting style is significantly greater than that of all the other styles, whereas the speaking rates of the other styles are similar to each other.

	Speaking Rate	
Speaker	Original	Newscasting
Michelle Obama	13.7	15.8
VOA	16.3	
Hillary Clinton	14.3	16.3
Blizzard 2017	15.0	15.7

Table 4.3. Speaking rate of the speakers in their original stylish utterances and synthetic newscasting utterances (of unseen news texts).

Effect on the Pitch

Fundamental frequencies (F0) contour shows how the pitch varies through a speech. An example F0 contour extracted by the Yin algorithm [6] is illustrated in Fig.4.2. The mean F0 is related to the speaking style. For instance, neutral speech is usually spoken with a flat pitch. Robinson et al. [38] modified the fundamental frequency of an utterance to convert it from a neutral speech to an emotional speech.

Synthetic utterances were generated by our proposed TTS model using Michelle Obama's voice from the held-out unseen texts of each style in that style and neutral style. The mean F0 of each pair of generated utterances were computed and compared in Fig. 4.3 – 4.5, and their average difference in each stylish test set is summarized in Table 4.4.

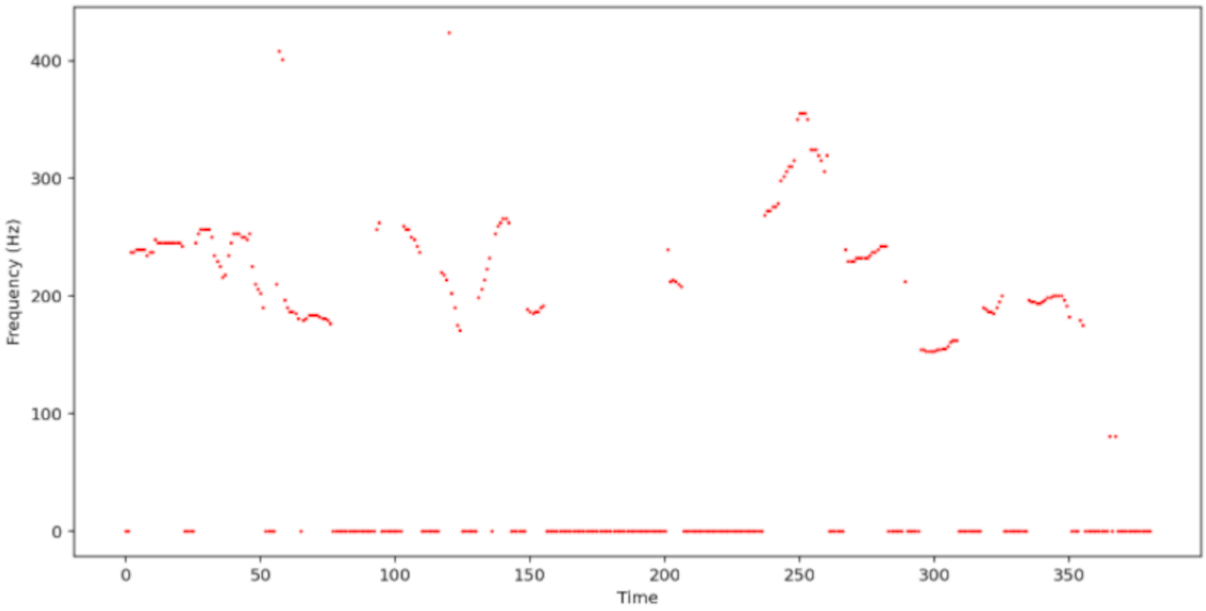

Figure 4.2. An example of Fundamental frequencies (F0) contour.

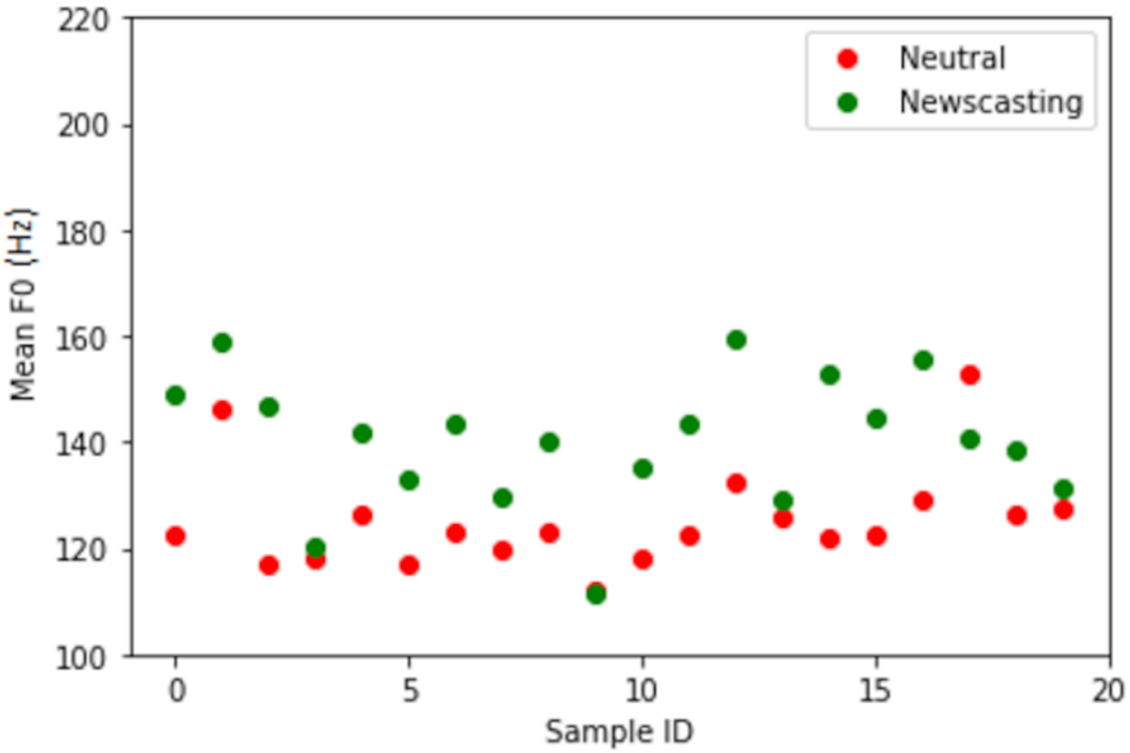

Figure 4.3. Mean F0 comparison: newscasting vs. neutral style.

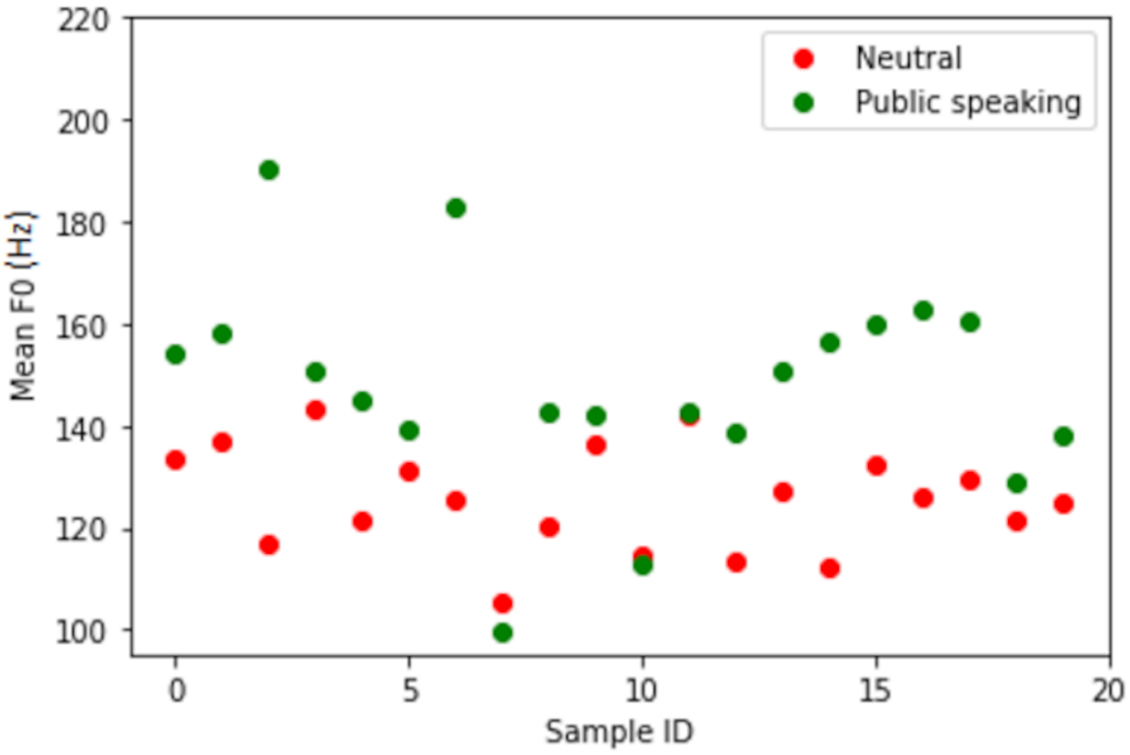

Figure 4.4. Mean F0 comparison: public speaking vs. neutral style.

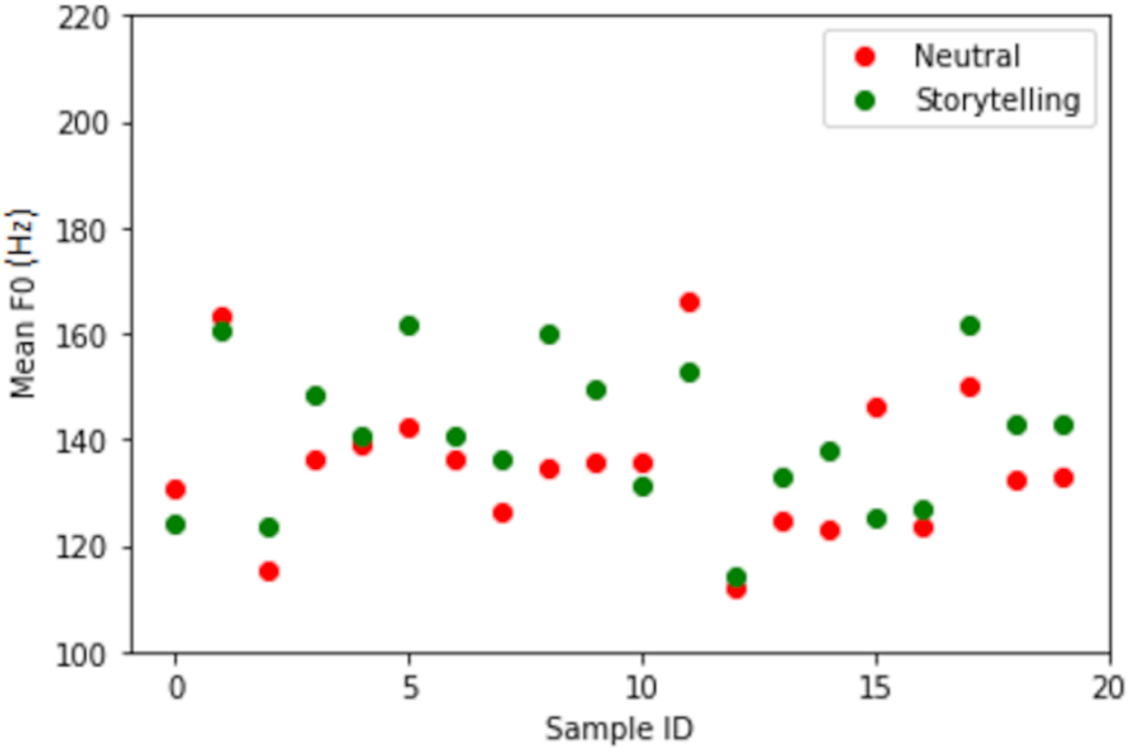

Figure 4.5. Mean F0 comparison: storytelling vs. neutral style.

We observe that the mean F0 is higher in all the other three styles than in neutral speech even though the utterances were all generated from the same speaker, Michelle Obama. Among the three non-neutral styles, public speech has the highest mean F0, while storytelling speech has the lowest mean F0. The same trend is found on the original datasets of the four styles: public speech $>$ newscasting speech $>$ storytelling speech $>$ neutral speech although their texts are different. This suggests that the text-predicted scenario-based GST module could capture the implicit property of the speech spoken in different scenarios.

Scenario	Newscasting	Public speaking	Storytelling
F0 absolute change	+16.3 Hz	+22.8 Hz	+10.1 Hz

Table 4.4. Average absolute increment of mean F0 when the same text is spoken with a scenario style vs. spoken with a neutral voice.

4.4.5 Subjective Evaluation

We performed a subjective evaluation on utterances synthesized by the proposed TTS model and the baseline TTS model. For each scenario, we used the neutral speaker's voice to generate utterances of 15 sentences in her neutral and scenario-specific style. For each sentence, two synthesized utterances from the two TTS models were played, and the listeners were asked which one they preferred for the desired scenario. The listeners also rated the naturalness and intelligibility of the synthesized utterances on an MOS scale from 1 to 5. Fifteen listeners were recruited to evaluate each scenario.

As discussed in Section 3.1.3, there are two options for data augmentation: (1) data augmentation only for the neutral speaker, Michelle Obama, or (2) data augmentation for all speakers. The preference results and MOS on naturalness and intelligibility of synthesized speech produced by option (1) are shown in Table 4.5 and Table 4.6, whereas the results of option (2) are shown in Table 4.7 and Table 4.8.

From Table 4.5 and Table 4.7, we find that generating augmented data with all speakers in various styles may cause more confusion as there are more counts of "no preference", but there are more subjects preferring our proposed model over the baseline model for

style transfer to newscasting style (by 60%), public speaking style (by 46%), and storytelling style (by 41%). The effect is particularly strong for the style transfer to newscasting speech. There can be two reasons for that: (a) data augmentation option 2 generates three times more augmented data for training our TTS models, resulting in more training samples for each scenario, and hence a better model; (b) even though our ultimate goal is only to generate synthesized speech from one neutral speaker, Michelle Obama in our case, in other three non-neutral styles, data augmentation option 2 renders our model training method a multi-task learning (MTL) method which tries to learn the conversion between any two of the four speaking styles. This helps the model to gain a better understanding of the subtle differences of all speaking styles.

Scenario	Proposed Model	Baseline Model	No Preference
Newscasting	69%	24%	7%
Public speaking	66%	27%	7%
Storytelling	68%	24%	8%

Table 4.5. Preference test results with data augmentation only for the neutral speaker.

Scenario	Naturalness	Intelligibility
Newscasting	3.79±0.15	3.95 ±0.13
Public speaking	3.56±0.10	3.61±0.09
Storytelling	3.74±0.10	3.71±0.12

Table 4.6. MOS results with data augmentation only for the neutral speaker at 95% confidence level.

Scenario	Proposed Model	Baseline Model	No Preference
Newscasting	74%	14%	12%
Public speaking	67%	21%	12%
Storytelling	69%	28%	3%

Table 4.7. Preference test results with data augmentation for all speakers.

Table 4.6 and Table 4.8 give the MOS results on the naturalness and intelligibility of the generated stylish utterances; they are all acceptable. We observe public speaking

Scenario	Naturalness	Intelligibility
Newscasting	3.79±0.08	3.99 ±0.08
Public speaking	3.33±0.11	3.67±0.08
Storytelling	3.80±0.11	3.92±0.12

Table 4.8. MOS results with data augmentation for all speakers at 95% confidence level.

speeches have a lower naturalness and intelligibility compared to newscasting and storytelling speeches. A plausible explanation is that the public speech data from Hillary Clinton were obtained from YouTube which has lower sound quality. Overall speech synthesized by our TTS model has a naturalness that is comparable to the reported audio signal quality of the voice-cloning-augmentation TTS model in [14].

4.5 Accent-beautified TTS

4.5.1 Dataset

We used the LJS corpus [16], which consists of roughly 21 hours of speech from a native English female speaker, and the L2-ARCTIC [61], which is a non-native English speech corpus. We selected the female Arabic speaker ZHAA, the female Chinese speaker LXC and the male Korean speaker YKWK for the experiments. Each L2 speaker recorded 1132 utterances, which were equivalent to roughly 1 hour of data. We used the first 1032 utterances for model training, the following 50 utterances for model validation, and selected 25 utterances from the remaining data for testing. The split is the same as used by a recent AC model [26], which gives state-of-the-art AC result on the L2-ARCTIC corpus.

The model optimizes the GSTs to capture the prosody of native and non-native speech from the training audio. The native speech from a single native speaker is ample for the GSTs learning purpose. For all intents and purposes, the native speaker provides the "golden standard" for native speech, while the L2 speaker imitates the golden standard's prosody to reduce his/her accent by conditioning on the native accent weighted GSTs.

4.5.2 Experimental Setup

We trained the ASR model with pairs of LJS's ground truth mel-spectrogram and phoneme sequence for 300 epochs. The ASR model was then frozen in subsequent training. We trained the TTS module with LJS speeches for 150 epochs. The resulting model serves as the pre-trained model in our system.

Then we experimented with four different speaker adaptation schemes by fine-tuning the pre-trained model on the speeches of individual L2 speakers. To evaluate the effect of **ASR** and **DA**, we performed an ablation study as follows:

1. **Fine-tuning (FT)**: For 100 epochs with only the TTS loss.
2. **FT + ASR**: For 100 epochs with only the TTS and ASR losses.
3. **FT + DA**: For 100 epochs with only the TTS and DA losses. Since each L2 speaker had fewer speech data than the native speaker's, the speech data of an L2 speaker were duplicated in each epoch to around the same amount (around 21 hours) as the native speaker's to tackle the data imbalance issue. We trained the model for 40 epochs, and then trained with on-the-fly augmented data for another 60 epochs.
4. **FT + ASR + DA**: We fine-tuned the model on the speeches of individual L2 speakers and LJS speaker with the TTS, ASR and DA losses. The training procedure is the same as that used in **FT + DA**.

4.5.3 Baseline Model

For evaluation, we compared the synthesized audios against those from a state-of-the-art AC model [26]. The AC model detail could be referred to Section 2.4.

4.5.4 Objective Evaluation

We evaluated the effectiveness of the proposed accent-beautified TTS method in two aspects: (a) pronunciation accuracy in terms of WER by running ASR on the synthesized L2 speeches, and (2) voice similarity decided by speaker verification.

An ASR model as a proxy of a native listener

We used the Deep speech model v0.9.3 released by Mozilla[7] as a proxy of a native English listener. The model was trained with thousands of hours of speech from multiple corpora. It achieved a 7.06% WER on the LibriSpeech clean test corpus. The model had a bias towards US accents.

From Table 4.9, we observe the **FT + ASR + DA** scheme reduces WER dramatically compared to the WER on the original L2 utterances and **FT** scheme. By comparing the schemes of (i) **FT + DA** against **FT** and (ii) **FT + ASR + DA** against **FT + ASR**, it shows that the DA loss term improves the clarity by 3 to 8%. In the following evaluations, we use the samples generated from the model of **FT + ASR +DA**.

	Speaker		
Utterances	ZHAA	LXC	YKWK
Original L2	26.5%	52.2%	46.9%
FT	23.1%	43.4%	38.8%
FT + DA	20.5%	35.4%	36.6%
FT + ASR	16.1%	19.9%	18.4%
FT + ASR + DA	**14.4%**	**15.2%**	**14.1%**
Samples of [26]	18.3%	NA	NA

Table 4.9. Word error rate (WER) of the original L2 and synthesized utterances.

Voice similarity measure

We applied the Resemblyzer model[8] to extract a speaker embedding from an utterance. Resemblyzer is a speaker verification system that was trained on speech data from 8K speakers. The model is an implementation of speaker embedding trained under the generalized end-to-end loss [50]. The Equal error rate (EER) is an indicator regarding the overall performance of a biometric system. It refers to the condition that the proportion of false acceptances is equal to the proportion of false rejections. The lower the EER is,

[7]https://github.com/mozilla/DeepSpeech/releases

[8]https://github.com/resemble-ai/Resemblyzer

the higher accuracy of the system has. Resemblyzer achieved around 4% on 9 enrollment utterances.

We retrieved the speaker embedding for each utterance in the test set. We compared the synthesized speech to the original L2 speech (of the same text) by computing the cosine similarity of the two embeddings as the similarity measure. We observe from Table 4.10 that the fine-tuning **FT** scheme gives a high speaker's voice similarity. Our **FT + ASR +DA** scheme gives a slightly worse speaker's voice similarity than **FT**, but it gives a much higher speaker's voice similarity than the model in [26].

Model	FT	FT + DA	FT + ASR	FT + ASR + DA	Samples of [26]
Score	0.861	0.857	0.831	0.819	0.738

Table 4.10. Voice similarity of the synthesized speech of ZHAA.

4.5.5 Subjective Evaluation

We performed a subjective evaluation on the (a) utterances synthesized by our proposed TTS model for speaker ZHAA. We compared them with the (b) original L2 utterances by ZHAA, (c) synthesized utterances by the native LJS speaker and (d) speech samples (of the same text) from [26]. Some audio samples are available on this webpage[9]. We focused on naturalness, voice similarity and accentedness. We recruited 15 listeners who declared themselves as native American English speakers. We summarized the results in Table 4.11 and Table 4.12.

Speaker	Proposed Model	Samples of [26]	No Preference
ZHAA	55%	26%	19%

Table 4.11. Preference test results on voice similarity.

For voice similarity evaluation, the listeners were asked to select the synthesized utterances that sounded most similar to the original samples of ZHAA that spoke the same content, and they were also given the choice of no preference. The result of the prefer-

[9]https://raymond00000.github.io/attsdemo.html

Audio	Naturalness	Accentedness
Original L2 utterances	3.74±0.08	6.38±0.18
Native TTS	3.56±0.13	4.66±0.18
Proposed TTS	3.37±0.11	5.82±0.23
Samples of [26]	3.42±0.08	5.31±0.18

Table 4.12. MOS results at 95% confidence level on naturalness and accentedness.

ence test confirmed our objective analysis that our model gave a higher speaker's voice similarity than [26].

Accentedness refers to the perceived strength of a non-native accent. We followed the MOS setting of [60] for its evaluation, in which the MOS scaled from 1 to 9, where 1 means no foreign accent and 9 means a very strong foreign accent. The result showed that our model could reduce the foreign accent compared to the original samples but not as good as [26].

Regarding the naturalness of the utterances, we used the MOS scale of 1 to 5. The original samples give the highest score as expected while our proposed model gives a comparable performance as [26].

CHAPTER 5

CONCLUSION AND FUTURE WORK

To the best of our knowledge, our proposed method is the first method that builds a multi-style text-to-speech model from a set of single-style disjoint datasets, each spoken by a different speaker. This is made possible by generating augmented speech data for the imitating speaker, conditioning on the style embedding extracted from target utterance and using a loss function over the alignment matrices (from the attention module in the model) of the imitating speaker and the original speaker. The trained model is a customized and expressive TTS model.

The backbone of our approach does not depend on any external ASR or VC model; it is a single model training approach. The data augmentation is done on-the-fly to utilize the latest model parameters for the data generation. Another benefit is our on-the-fly augmented unpaired data share some embedding variables with the real paired data hence the GPU RAM could be potentially optimized for a larger mini-batch size during training.

We substantiated the effectiveness of our proposed method in two TTS applications. For the scenario-based TTS, objective evaluation on the rhythm and pitch profile of the synthesized stylish speeches shows that our proposed model successfully performs the style transfer. Subjective evaluation also shows that stylish speech generated by our proposed model is overwhelmingly preferred over the baseline model that was trained to generate neutral speech.

For the accent-beautified TTS, we extended the proposed model with a phoneme recognition module. We built a TTS model based on a native speaker corpus and a (relatively) small amount of L2 utterances from a non-native speaker. The native speaker corpus acts as a guide to reduce the foreign accent of the synthesized L2 speech of the non-native speaker. Both subjective and objective evaluations show that the model gives a high speaker similarity because of the joint training of the speaker embedding. The proposed alignment loss for on-the-fly augmented data together with ASR loss also improves

the clarity of the synthesized speech. In terms of accentedness, subjective MOS shows our model could reduce the foreign accent of the speech with comparable naturalness of a state-of-the-art model [26].

Regarding the limitation, our proposed approach focuses on one-to-one speech imitation. For instance, a person tries to imitate Steve Jobs's presentation speech to improve his/her public speaking skill. Another situation is the person tries to imitate two speakers, Barack Obama and George Walker Bush, to improve his/her public speaking speech. In the latter case, the proposed model would require multiple speakers of the same scenario spoken in a very similar speaking style.

In the future, we could gather audio data in other scenarios, style transfer across opposite gender speakers, or even a multilingual speech dataset to further evaluate the style transfer capability of the proposed TTS system in challenging circumstances. In addition, contextual word embedding (CWE) considers the sentence context in word embedding calculation. In the field of natural language processing, CWE was used to perform the part-of-speech tagging task. CWE derived from the input sentence was shown to be beneficial to the TTS in speaking style modeling [36]. We could include CWE as well to validate the benefit. Last but not least, we could investigate replacing the speaker embedding with a very well-trained speaker encoder in the pursuit of a zero-shot stylish speech synthesis of a common person.

As a final closing remark: TTS training usually requires a large training corpus. Our experiment on found data from the web shows it is feasible to mine audio data from audiobooks and YouTube for stylish TTS training. We also demonstrate the proposed novel data augmentation method enables a TTS style transfer using disjoint datasets.

REFERENCES

[1] Sercan Ö Arık, Mike Chrzanowski, Adam Coates, Gregory Diamos, Andrew Gibiansky, Yongguo Kang, Xian Li, John Miller, Andrew Ng, Jonathan Raiman, et al. Deep Voice: Real-time neural text-to-speech. In *International Conference on Machine Learning*, pages 195–204. PMLR, 2017.

[2] Jae-Sung Bae, Hanbin Bae, Young-Sun Joo, Junmo Lee, Gyeong-Hoon Lee, and Hoon-Young Cho. Speaking speed control of end-to-end speech synthesis using sentence-level conditioning. *arXiv preprint arXiv:2007.15281*, 2020.

[3] Mark Beutnagel, Alistair Conkie, and Ann K Syrdal. Diphone synthesis using unit selection. In *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.

[4] Blogcast. Create a podcast without recording. `https://blogcast.host/`. Accessed: 2021-10-21.

[5] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964. IEEE, 2016.

[6] Alain De Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.

[7] Gagandeep Singh. Statistical parametric speech synthesis with focus on LDM based TTS. `http://www.cs.toronto.edu/~gagandeep/files/spss.pdf`. Accessed: 2021-10-21.

[8] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376, 2006.

[9] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, et al. Deep Speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[10] Gustav Eje Henter, Simon King, Thomas Merritt, and Gilles Degottex. Analysing shortcomings of statistical parametric speech synthesis. *arXiv preprint arXiv:1807.10941*, 2018.

[11] Geoffrey E Hinton, Alex Krizhevsky, and Ilya Sutskever. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1106–1114, 2012.

[12] Qiong Hu, Tobias Bleisch, Petko Petkov, Tuomo Raitio, Erik Marchi, and Varun Lakshminarasimhan. Whispered and Lombard neural speech synthesis. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 454–461. IEEE, 2021.

[13] Andrew J Hunt and Alan W Black. Unit selection in a concatenative speech synthesis system using a large speech database. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, volume 1, pages 373–376. IEEE, 1996.

[14] Goeric Huybrechts, Thomas Merritt, Giulia Comini, Bartek Perz, Raahil Shah, and Jaime Lorenzo-Trueba. Low-resource expressive text-to-speech using data augmentation. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6593–6597. IEEE, 2021.

[15] Washington Irving. The Legend of Sleepy Hollow. `https://play.google.com/store/audiobooks/details/Washington_Irving_The_Legend_of_Sleepy_Hollow?id=AQAAAEDszTONzM`. Accessed: 2021-10-21.

[16] Keith Ito and Linda Johnson. The LJ Speech dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[17] Maysa Jaber and Riyad F Hussein. Native speakers' perception of non-native English speech. *English Language Teaching*, 4(4):77–87, 2011.

[18] Sri Karlapati, Alexis Moinet, Arnaud Joly, Viacheslav Klimkov, Daniel Sáez-Trigueros, and Thomas Drugman. Copycat: Many-to-many fine-grained prosody transfer for neural text-to-speech. *arXiv preprint arXiv:2004.14617*, 2020.

[19] Hideki Kawahara. STRAIGHT, exploitation of the other aspect of VOCODER: Perceptually isomorphic decomposition of speech sounds. *Acoustical Science and Technology*, 27(6):349–353, 2006.

[20] Kartik Khandelwal, Preethi Jyothi, Abhijeet Awasthi, and Sunita Sarawagi. Black-box adaptation of ASR for accented speech. *arXiv preprint arXiv:2006.13519*, 2020.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018.

[23] Debi Laplante and Nalini Ambady. On how things are said: Voice tone, voice intensity, verbal content, and perceptions of politeness. *Journal of Language and Social Psychology*, 22(4):434–441, 2003.

[24] Younggun Lee, Azam Rabiee, and Soo-Young Lee. Emotional end-to-end neural speech synthesizer. *arXiv preprint arXiv:1711.05447*, 2017.

[25] Stan Z. Li and Anil Jain, editors. *Fundamental Frequency, Pitch, F0*, pages 592–592. Springer US, Boston, MA, 2009.

[26] Wenjie Li, Benlai Tang, Xiang Yin, Yushi Zhao, Wei Li, Kang Wang, Hao Huang, Yuxuan Wang, and Zejun Ma. Improving accent conversion with reference encoder and end-to-end text-to-speech. *arXiv preprint arXiv:2005.09271*, 2020.

[27] Da-Rong Liu, Chi-Yu Yang, Szu-Lin Wu, and Hung-Yi Lee. Improving unsupervised style transfer in end-to-end speech synthesis with end-to-end speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 640–647. IEEE, 2018.

[28] Songxiang Liu, Disong Wang, Yuewen Cao, Lifa Sun, Xixin Wu, Shiyin Kang, Zhiyong Wu, Xunying Liu, Dan Su, Dong Yu, et al. End-to-end accent conversion without

using native utterances. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6289–6293. IEEE, 2020.

[29] Kenji Matsui, Stephen D Pearson, Kazue Hata, and Takahiro Kamai. Improving naturalness in text-to-speech synthesis using natural glottal source. In *Acoustics, Speech, and Signal Processing, IEEE International Conference on*, pages 769–772. IEEE Computer Society, 1991.

[30] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger. Montreal Forced Aligner: Trainable text-speech alignment using Kaldi. In *Interspeech*, volume 2017, pages 498–502, 2017.

[31] Narro. Turn your articles into a podcast. `https://www.narro.co/`. Accessed: 2021-10-21.

[32] Noël Nguyen, Marc Sato, Marie Postma-Nilsenová, Jennifer Pardo, and Molly Babel. Speech imitation: the cognitive underpinnings of adaptive vocal behaviour, 2013.

[33] JO Onaolapo, FE Idachaba, JA Badejo, T Odu, and OI Adu. A simplified overview of text-to-speech synthesis. In *Proceedings of the World Congress on Engineering 2014 Vol I*, 2014.

[34] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel WaveNet: Fast high-fidelity speech synthesis. In *International Conference on Machine Learning*, pages 3918–3926. PMLR, 2018.

[35] Dipjyoti Paul, Muhammed PV Shifas, Yannis Pantazis, and Yannis Stylianou. Enhancing speech intelligibility in text-to-speech synthesis using speaking style conversion. *arXiv preprint arXiv:2008.05809*, 2020.

[36] Nishant Prateek, Mateusz Łajszczak, Roberto Barra-Chicote, Thomas Drugman, Jaime Lorenzo-Trueba, Thomas Merritt, Srikanth Ronanki, and Trevor Wood. In other news: A bi-style text-to-speech model for synthesizing newscaster voice with limited data. *arXiv preprint arXiv:1904.02790*, 2019.

[37] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. WaveGlow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[38] Carl Robinson, Nicolas Obin, and Axel Roebel. Sequence-to-sequence modelling of F0 for speech emotion conversion. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6830–6834. IEEE, 2019.

[39] Giovanni Segar. How to keep the audience engaged. `https://potentspeaking.com/keep-audience-engaged/`. Accessed: 2021-10-21.

[40] Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, and Rj Skerrv-Ryan. Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[41] Skype. How to use Skype Translator. `https://youtu.be/UnEbQtp_r3A`, 2015.

[42] Daisy Stanton, Yuxuan Wang, and RJ Skerry-Ryan. Predicting expressive speaking style from text in end-to-end speech synthesis. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 595–602. IEEE, 2018.

[43] Stanley Smith Stevens, John Volkmann, and Edwin Broomell Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[44] Chrispy Things. I asked Alexa and Google Home to tell me jokes. Try not to laugh! `https://www.youtube.com/watch?v=stV5VIsycAs`. Accessed: 2021-10-21.

[45] Noé Tits, Kevin El Haddad, and Thierry Dutoit. Exploring transfer learning for low resource emotional TTS. In *Proceedings of SAI Intelligent Systems Conference*, pages 52–60. Springer, 2019.

[46] Carnegie Mellon University. The CMU pronouncing dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`. Accessed: 2021-10-21.

[47] Rafael Valle, Jason Li, Ryan Prenger, and Bryan Catanzaro. Mellotron: Multispeaker expressive voice synthesis by conditioning on rhythm, pitch and global style tokens. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6189–6193. IEEE, 2020.

[48] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[49] Debora Vigliano, Kei Yoshimoto, and Elisa Pellegrino. A self-imitation technique for the improvement of prosody in L2 Italian. In *Proceedings of the 22nd Annual Meeting of the The Association for Natural Language Processing*, pages 1189–1192. IEEE, 2016.

[50] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4879–4883. IEEE, 2018.

[51] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al. Tacotron: Towards end-to-end speech synthesis. *arXiv preprint arXiv:1703.10135*, 2017.

[52] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A. Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR, 2018.

[53] Matt Whitehill, Shuang Ma, Daniel McDuff, and Yale Song. Multi-reference neural TTS stylization with adversarial cycle consistency. *arXiv preprint arXiv:1910.11958*, 2019.

[54] Wikipedia contributors. Stephen Hawking — Wikipedia, the free encyclopedia, 2021. [Online; accessed 20-October-2021].

[55] MacDonald Kirsten Yamagishi Junichi, Veaux Christophe. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit. `https://datashare.ed.ac.uk/handle/10283/3443/`, 2019.

[56] Heiga Zen, Viet Dang, Rob Clark, Yu Zhang, Ron J Weiss, Ye Jia, Zhifeng Chen, and Yonghui Wu. LibriTTS: A corpus derived from LibriSpeech for text-to-speech. *arXiv preprint arXiv:1904.02882*, 2019.

[57] Heiga Zen, Keiichi Tokuda, and Alan W Black. Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064, 2009.

[58] Hao Zhang, Richard Sproat, Axel H Ng, Felix Stahlberg, Xiaochang Peng, Kyle Gorman, and Brian Roark. Neural models of text normalization for speech applications. *Computational Linguistics*, 45(2):293–337, 2019.

[59] Yang Zhang, Liqun Deng, and Yasheng Wang. Unified Mandarin TTS front-end based on distilled BERT model. *arXiv preprint arXiv:2012.15404*, 2020.

[60] Guanlong Zhao, Shaojin Ding, and Ricardo Gutierrez-Osuna. Converting foreign accent speech without a reference. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2367–2381, 2021.

[61] Guanlong Zhao, Sinem Sonsaat, Alif O Silpachai, Ivana Lucic, Evgeny Chukharev-Hudilainen, John Levis, and Ricardo Gutierrez-Osuna. L2-ARCTIC: A non-native English speech corpus. *Perception Sensing Instrumentation Lab*, 2018.

[62] Edison Zhou. Preliminary exploration of machine learning using iFLYTEK TTS service to achieve online speech synthesis. `https://www.cnblogs.com/edisonchou/p/edc_machine_learning_xunfeicloud_online_tts_introduction.html`. Accessed: 2021-10-21.

[63] Tao Zhou, Yuan Dong, Dezhi Huang, Wu Liu, and Haila Wang. A three-stage text normalization strategy for Mandarin text-to-speech systems. In *2008 6th International Symposium on Chinese Spoken Language Processing*, pages 1–4. IEEE, 2008.

[64] Xiaolian Zhu, Yuchao Zhang, Shan Yang, Liumeng Xue, and Lei Xie. Pre-alignment guided attention for improving training efficiency and model stability in end-to-end speech synthesis. *IEEE Access*, 7:65955–65964, 2019.

[65] Yingke Zhu, Tom Ko, and Brian Mak. Mixup learning strategies for text-independent speaker verification. In *Interspeech*, pages 4345–4349, 2019.

List of Publications

Raymond Chung and Brian Mak, "On-the-fly Data Augmentation for Text-to-speech Style Transfer", ~~**accepted** in 2021~~ IEEE Automatic Speech Recognition and Understanding Workshop, 2021.

Raymond Chung and Brian Mak, "Accent-beautified Text-to-speech Synthesis with Data Augmentation and Phoneme Prediction Loss", **submitted** to 2022 IEEE International Conference on Acoustics, Speech and Signal Processing.